

Adobe LiveCycle Production Print ES2

Version 9.0

Manuel utilisateur

Rév. B



Adobe LiveCycle Production Print ES2 - Manuel utilisateur
Rév. B

Brevet américain n° 7,127,520

Site Web de StreamServe
http://www.streamserve.com

Adobe® LiveCycle® Production Print ES2

La licence de ce produit a été acquise auprès d'Adobe Systems Incorporated ou d'une société tierce agréée par Adobe.

Il s'agit d'un produit sous licence exploitant une technologie développée par StreamServe Inc.

L'utilisation de ce logiciel est soumise au contrat de licence utilisateur final (CLUF) d'Adobe Systems Incorporated.

Tous les services de maintenance et de support sont assurés par Adobe Systems Incorporated ou un tiers autorisé par Adobe.

Les conditions générales d'utilisation du logiciel sont décrites dans le contrat de licence (CLUF) accompagnant le produit fourni par Adobe Systems Incorporated.

Pour tout problème de licence, contactez Adobe Systems Incorporated.

Le support physique et la documentation d'installation contiennent et mentionnent des composants qui ne sont peut-être pas utilisables par ou avec votre licence Adobe LiveCycle Production Print ES2.

Coordonnées d'Adobe Systems Incorporated.

Pour obtenir des mises à jour de correctifs, des notes techniques et d'autres informations sur ce produit, contactez votre fournisseur Adobe Enterprise Support ou consultez la page

www.adobe.com/support/products/enterprise/index.html

Pour d'autres questions d'ordre général, reportez-vous à la page

http://www.adobe.com/aboutadobe/contact.html

Informations destinées aux développeurs

Accédez au site Web Adobe LiveCycle Developer Center, www.adobe.com/devnet/livecycle, pour obtenir les informations les plus récentes en matière de développement et continuer à vous former grâce aux nombreuses ressources à votre disposition (articles, tutoriels, exemples de codes, fichiers à télécharger et exemples d'applications).

Pour plus d'informations sur les ressources disponibles pour les développeurs, consultez la page www.adobe.com/enterprise/developer/main.html.

Table des matières

A propos de LiveCycle Production Print ES2

Adobe® LiveCycle® Production Print ES2 permet aux entreprises qui en ont besoin de créer des documents depuis leur centre de données dans leur environnement de production en générant de façon dynamique des documents personnalisés en sortie dans différents formats (par exemple, AFP, PostScript®, PDF ou PCL) à partir de données XML issues d'applications de base. LiveCycle Production Print ES2 permet d'étendre l'accès à LiveCycle® Designer ES2 et à LiveCycle ES2 Platform. Les utilisateurs peuvent ainsi produire des documents standardisés dans un environnement de conception unique pour toutes les communications sensibles de l'entreprise.

LiveCycle Production Print ES2 fournit des fonctionnalités d'impression de production pour l'ensemble des logiciels de la solution Adobe LiveCycle ES2. La solution est une solution intégrée StreamServe - Adobe où Adobe LiveCycle Designer ES2 est intégré dans l'environnement StreamServe.

Ce manuel contient des informations propres à LiveCycle Production Print ES2. Pour obtenir des informations d'ordre général sur StreamServe, voir la documentation standard correspondante. Ce guide s'adresse aux lecteurs ayant acquis une compréhension suffisante des concepts de base de StreamServe et qui ont idéalement suivi le cours StreamServe Persuasion - Notions de base.

StreamServe Persuasion

LiveCycle Production Print ES2 repose sur StreamServe Persuasion SP4, un logiciel d'EDP (Enterprise Document Presentment) développé par StreamServe. Les composants de StreamServe Persuasion sont les suivants :

- StreamServe Component Framework, qui inclut les logiciels sous-jacents permettant d'exécuter des applications StreamServe.
- StreamServer, le logiciel qui permet d'exécuter des applications StreamServer.
- Design Center, l'outil de conception le plus important dans StreamServe Persuasion. Dans Design Center, vous pouvez :
  - Créer des Projets StreamServe.
  - Configurer le mode de connexion à l'application source.
  - Identifier et extraire des données d'entrée.
  - Configurer le mode d'envoi des données de sortie sur des périphériques de sortie.

Documentation utilisateur

La documentation StreamServe standard est utilisée en parallèle avec la documentation propre à LiveCycle Production Print ES2.

Les documents propres à LiveCycle Production Print ES2 sont les suivants :

- Notes de version : ce document décrit les fonctionnalités nouvelles et modifiées.
- Manuel utilisateur : le présent document décrit les fonctionnalités de LiveCycle Production Print ES2.
- Manuel d'installation : ce document décrit la procédure d'installation, de mise à niveau et de vérification du logiciel LiveCycle Production Print ES2. Il répertorie également les plates-formes et logiciels pris en charge.

Projet exemple

Un Projet exemple est fourni lors de l'installation de StreamServe (voir *Projet exemple* à la page 57).

Introduction

Adobe® LiveCycle® Designer ES2 est l'outil de conception de formulaires développé par Adobe, intégré à l'environnement de conception StreamServe et utilisé comme outil Processus StreamServe. La sortie est générée dans divers formats. D'aspect, elle est strictement identique à la sortie générée à partir de LiveCycle Output ES2.

StreamServe Design Center peut utiliser Adobe LiveCycle Designer ES2 en tant que composant autonome ou en tant que composant intégré à LiveCycle Workbench ES2.

Vous pouvez fusionner un modèle de formulaire existant avec une instance de document XML ou vous pouvez vous-même créer un modèle de formulaire (avec des liaisons vers des formats de données autres que le format XML) dans StreamServe Design Center.

L'outil Processus pour LiveCycle Design ES2 vous permet d'effectuer les opérations suivantes :

- Importer un modèle de formulaire LiveCycle Designer ES2 existant
- Créer un modèle de formulaire LiveCycle Designer ES2
- Modifier un modèle de formulaire à l'aide de LiveCycle Designer ES2
- Exporter des fichiers d'aperçu et des schémas pour des formats autres que le format XML en vue d'une utilisation dans une application LiveCycle Designer ES2 autonome

Exécution sans Adobe LiveCycle Designer ES2

Si vous utilisez un modèle de formulaire existant, vous pouvez exécuter l'outil Processus même si LiveCycle Designer ES2 n'est pas installé. Toutefois, vous ne pourrez pas modifier le modèle de formulaire dans l'environnement de conception StreamServe.

Dans ce chapitre :

Modèles

Vous pouvez charger des modèles à partir d'un jeu de ressources dans le Projet Design Center ou à partir du référentiel LiveCycle ES2.

A partir d'un jeu de ressources Design Center

Les modèles chargés à partir d'un jeu de ressources dans le Projet Design Center sont associés au Processus de façon statique. Ils sont chargés au moment du démarrage.

A partir du référentiel LiveCycle

Les modèles chargés à partir du référentiel LiveCycle peuvent être traités de différentes façons :

- Chargement lors de la conception et stockage dans un jeu de ressources Design Center.
  Design Center se connecte au référentiel LiveCycle et charge les modèles dans le jeu de ressources Design Center. Vous pouvez les mettre à jour en les chargeant à partir du référentiel LiveCycle.
- Chargement dynamique durant l'exécution via une connexion au référentiel LiveCycle au moyen de variables StreamServe ou d'expressions SOM. Il est possible de charger ces modèles une seule fois pour chaque job ou une seule fois pour chaque enregistrement traité.

Pour plus d'informations sur l'accès aux modèles et sur leur chargement dynamique, voir *Utilisation de modèles* à la page 23.

Remarques relatives aux performances

En règle générale, le chargement dynamique des modèles a un impact négatif sur les performances. Elles dépendent essentiellement du système de fichiers, des entrées/sorties de fichier et de la capacité du réseau. Pour obtenir des performances optimales, assurez-vous que le Processus peut accéder aux modèles le plus rapidement possible.

Pour améliorer les performances lorsque vous utilisez des modèles dynamiques, vous pouvez activer leur mise en mémoire cache. Voir *Mise en mémoire cache de modèles* à la page 12.

Dépendances – Fragments et images

Les fichiers XDP avec des dépendances (références externes à des fragments et des images) sont pris en charge. Un fichier référencé n'est pas incorporé dans le fichier XDP principal. Lorsque vous importez un modèle avec des dépendances, l'outil Processus résout les références et importe les ressources requises.

StreamServer peut traiter des dépendances à partir du système de fichiers local, du réseau local, ou de ressources HTTP et FTP.

Lorsque vous importez un fichier LCA (LiveCycle Archive), le fichier XDP principal et toutes ses dépendances sont inclus dans le package LCA importé.

Les modèles et les fichiers associés peuvent être importés du référentiel Adobe LiveCycle vers un jeu de ressources dans le Projet Design Center. Voir *Accès au référentiel Adobe LiveCycle ES2* à la page 32.

Note : Les dépendances importées dans l'outil Processus ne seront pas mises à jour automatiquement en cas de mise à jour des ressources importées dans leur emplacement source.
Si des dépendances sont modifiées à l'emplacement source, vous devez les mettre à jour dans l'outil Processus.

Post-traitement

Le post-traitement dans LiveCycle Production Print ES2 génère une sortie de la même façon que l'outil StreamServe PageOUT.

Lorsque vous utilisez un modèle de formulaire existant avec un jeu de données XML contenant plusieurs enregistrements, le post-traitement ne traite pas chaque enregistrement comme un Message distinct. Au contraire, il traite les enregistrements par lot au sein d'un Processus.

Exécution de scripts avant et après chaque enregistrement

Vous pouvez exécuter des scripts Avant et Après Processus, avant et après chaque enregistrement. Vous devez pour cela sélectionner une option dans la boîte de dialogue Paramètres. Voir *Boîte de dialogue Paramètres* à la page 84.

Le tout premier script Avant Processus et le tout dernier script Après Processus sont exécutés dans les phases de prétraitement et de traitement. Les autres scripts Avant et Après Processus sont exécutés dans la phase de traitement uniquement. Il est important de tenir compte de cette particularité lors de la configuration des scripts.

Voir *Utilisation de scripts Avant et Après Processus* à la page 104.

Regroupement en documents logiques – déclencheur de Document

En règle générale, dans LiveCycle Production Print ES2, chaque enregistrement est automatiquement mappé avec un document.

Pour fractionner et regrouper les données de sortie des Processus connectés au connecteur de sortie, vous pouvez utiliser le déclencheur de Document. Ceci peut par exemple s'avérer utile si tous les documents portant le même numéro de client dans le job d'entrée doivent être inclus dans le même document. Le déclencheur de Document est défini dans la boîte de dialogue Paramètres du Connecteur de sortie de Runtime (voir la *documentation sur StreamServer Design Center*).

Vous pouvez utiliser le déclencheur de Document pour les modes de sortie Document et Job.

Utilisation du déclencheur automatique de Document

Pour conserver le mappage automatique de chaque enregistrement avec un document, utilisez le paramètre **Déclencheur automatique de Document**. Si vous le sélectionnez, la variable du déclencheur de Document est désactivée.

Ce paramètre est sélectionné par défaut pour les Projets mis à niveau à partir de versions antérieures à LiveCycle Production Print ES2.

Voir *Boîte de dialogue Paramètres de Processus de Runtime - onglet Général* à la page 96.

Pour activer le déclencheur automatique de Document

1. Dans la vue Runtime, cliquez avec le bouton droit de la souris sur le Processus LiveCycle Designer ES2 et sélectionnez **Paramètres**. La boîte de dialogue Paramètres de Processus de Runtime s'ouvre.
2. Cliquez sur l'onglet **Général**.
3. Sélectionnez l'option **Déclencheur automatique de Document**.
   La **variable du déclencheur de Document** (définie dans la boîte de dialogue Paramètres du Connecteur de sortie de Runtime, onglet Déclencheur de Document) est maintenant désactivée et chaque enregistrement sera automatiquement mappé avec un document. Il n'y aura pas de regroupement des données de sortie en documents logiques.

Remarques relatives aux performances

Mise en mémoire cache des données de polices

Pour améliorer les performances, vous pouvez utiliser l'argument de démarrage `preloadmorefontdata` qui active la mise en mémoire cache des données de polices supplémentaires au démarrage. Cet argument permet d'accélérer l'exécution, mais le démarrage est plus lent et la mémoire consommée plus importante.

Par défaut, ces données ne sont pas copiées en mémoire cache.

Voir la documentation *StreamServe* sur les *arguments de démarrage*.

Mise en mémoire cache de modèles

Le chargement et le déchargement de modèles de formulaires peut nuire aux performances. Si vous utilisez un même modèle plusieurs fois, comme c'est le cas avec les modèles dynamiques, les performances sont sensiblement meilleures en cas de mise en mémoire cache.

Vous activez/désactivez la mise en mémoire cache des modèles dans la boîte de dialogue Paramètres. Voir *Boîte de dialogue Paramètres* à la page 84.

Si le modèle de formulaire d'origine a été modifié, il sera automatiquement rechargé dans la mémoire cache. L'horodatage du fichier de modèle d'origine et du fichier mis en mémoire cache permet de déterminer si le fichier d'origine a été modifié. Cela ne s'applique qu'au fichier de modèle principal. Les fragments modifiés ne sont pas rechargés.

Vous pouvez définir la taille du cache de modèles dans Control Center ainsi que par le biais de la ligne de commande. La taille du cache est exprimée en Ko et elle remplace la taille du cache définie par défaut.

Par défaut, la taille du cache est de 10 240 Ko.

Control Center

L'option **Taille du cache** est définie dans la vue Propriétés.

Ligne de commande

```
-maxcachesize <valeur>
```

Formats de sortie pris en charge

Le processeur XFA prend en charge les pilotes et les formats de sortie suivants :

Format de sortie	Pilote LiveCycle Production Print ES2
AFP – Advanced Function Presentation	AFP
IJPDS – Ink Jet Printer Data Stream	IJPDS
PDF – Portable Document Format	PDF
PDF balisé	PDF
PDF/A 1a	PDF
PDF/A 1b	PDF
PCL – Printer Control Language	PCL5
P.S. – Postscript	Postscript
TIFF – Tagged Image File Format	TIFF
ZPL II – Zebra Programming Language	Zebra ZPLII
FP/DP – Intermec Fingerprint/ Direct Protocol	Intermec FP/DP
IPL – Intermec Printer Language	Intermec IPL
PGL/IGP – Intelligent Graphics Printing/ Printronix Graphics Language	Printronix PGL/IGP
Toshiba TEC	TEC
API Windows Print	Pilote Windows (StreamServe 3.x)

Les autres formats de sortie StreamServe Persuasion n'ont pas été testés, mais peuvent toutefois fonctionner avec certaines limites quant aux polices et au rendu des objets.

Reportez-vous à la documentation StreamServe standard pour plus d'informations sur les formats de sortie pris en charge par StreamServe Persuasion SP4.

Pilote Windows

Vous pouvez créer un fichier de configuration de pilote Adobe LiveCycle Production Print ES2 (`.drs`) pour le pilote Windows d'Adobe LiveCycle Production Print ES2, à utiliser avec des pilotes d'imprimante tiers spécifiques. Voir la documentation sur les *outils de gestion des pilotes de périphériques* StreamServe.

Césure

LiveCycle Production Print ES2 prend en charge la césure, de façon similaire à Output ES2 et Forms ES2.

Pour plus d'informations sur les limitations connues en matière de césure, voir les *notes de version sur Adobe LiveCycle Production Print ES2*.

Utilisation d'un modèle de formulaire existant

Ce chapitre décrit comment utiliser un modèle de formulaire existant à des fins d'impression de production. L'instance de document XML est fusionnée avec un modèle de formulaire.

Principe de fonctionnement dans StreamServer

1. L'Evénement XMLIN détecte l'instance de document XML et déclenche le Processus.
2. Ce dernier fusionne le contenu de l'instance de document XML d'entrée avec le modèle de formulaire et génère des données de sortie structurées.
3. Les données de sortie obtenues sont envoyées au pilote et en post-traitement pour être mises en forme.
4. La sortie mise en forme est envoyée à sa destination.

Dans ce chapitre :

Configuration du Projet

Prérequis :

- Une instance de document XML servant de données d'entrée.
- Un modèle de formulaire, contenant des liaisons de données vers l'instance de document XML.
- Un Projet StreamServe ouvert dans Design Center.

Pour ajouter le modèle de formulaire en tant que ressource

1. Dans le navigateur de Projet de Design Center, double-cliquez sur le jeu de ressources. La vue Jeu de ressources s'ouvre.
2. Cliquez avec le bouton droit de la souris sur le jeu de ressources, sélectionnez **Importer** et accédez au fichier à importer en tant que ressource pour le sélectionner. La ressource est créée et ajoutée au jeu de ressources.

Pour créer le Message

1. Sélectionnez **Fichier > Nouveau > Message**. Un Message vide est créé.
2. Renommez le Message.

Pour créer l'Evénement

1. Cliquez avec le bouton droit de la souris sur la vue Message et sélectionnez **Ajouter un Evénement > XMLIN**. Un nouvel Evénement est ajouté à la vue Message.
2. Renommez l'Evénement.
3. Cliquez avec le bouton droit de la souris sur cet Evénement et sélectionnez **Ouvrir**. L'outil Evénement s'ouvre.
4. Ouvrez l'instance de document XML en tant qu'exemple en sélectionnant **Fichier > Ouvrir un fichier exemple.** La boîte de dialogue Sélectionner une ressource s'ouvre.
5. Recherchez et sélectionnez le fichier. Le fichier s'affiche dans la vue des fichiers exemples XMLIN.
6. Utilisez l'Outil Identifiant pour créer un identifiant qui déclenchera l'Evénement. Voir la documentation sur *StreamServe XMLIN*.
7. Enregistrez les données et quittez l'outil Evénement.

Pour créer le Processus

1. Cliquez avec le bouton droit de la souris dans la vue Message et sélectionnez **Ajouter un Processus > Adobe LiveCycle Designer ES2**. Un nouveau Processus est ajouté à la vue Message.
2. Renommez le Processus.

3 Cliquez avec le bouton droit de la souris sur ce Processus et sélectionnez **Ouvrir**. L'outil Processus s'ouvre et la boîte de dialogue Paramètres s'affiche. Voir *Boîte de dialogue Paramètres* à la page 84.

4 Cliquez sur **Charger l'interface utilisateur de LiveCycle Designer au démarrage** et sur **OK**. Notez que cette étape est facultative car vous n'avez peut-être pas installé LiveCycle Designer.

   LiveCycle Designer est lancé et affiche une zone de dessin vide. Dans la vue Données, la connexion de données est vide car aucun Message n'est connecté.

5 Importez le modèle de formulaire. Pour ce faire, sélectionnez **Fichier > Ouvrir un modèle**. La boîte de dialogue Sélection du modèle s'ouvre.

6 Cochez la case **A partir du jeu de ressources Design Center** et sélectionnez le fichier du modèle de formulaire. LiveCycle Designer est lancé et le modèle de formulaire est affiché dans la zone de dessin.

7 Enregistrez les données et quittez l'outil Processus.

Pour finaliser le Projet

1 Configurez la Plate-forme.

2 Configurez le Runtime.

3 Exportez le Projet.

Note : Lorsque vous définissez les paramètres d'Evénement de Runtime, vous pouvez utiliser l'option **Ignorer les données restantes** pour améliorer les performances. Cette option permet d'ignorer les données restantes lorsque l'identifiant de déclencheur est trouvé. Voir le chapitre sur la configuration du Runtime dans la documentation sur *StreamServe Design Center.*

L'étape suivante consiste à déployer et à exécuter le Projet (voir *Déploiement et exécution* à la page 18).

Déploiement et exécution

Pour déployer le Projet

Vous déployez un Projet vers une application StreamServer dans Control Center :

1. Créez l'application StreamServer vers laquelle vous voulez déployer le Projet.
2. Déployez le Projet vers cette application.

Voir la documentation sur *StreamServe Control Center*.

Pour exécuter l'application StreamServer

Lancez l'application StreamServer dans Control Center en cliquant avec le bouton droit de la souris sur le noeud de l'application et sélectionnez **Démarrer**.

De même, vous arrêtez et redéployez des applications StreamServer dans Control Center. Voir la documentation sur *StreamServe Control Center*.

Utilisation d'un modèle de formulaire défini entièrement par l'utilisateur

Ce chapitre explique comment :

- Créer un modèle de formulaire en définissant tout vous-même et en utilisant un Message StreamServe pour les liaisons de champs et de blocs.
- Activer le modèle pour une impression de production.

Principe de fonctionnement dans StreamServer

1. L'Evénement détecte le fichier d'entrée, extrait les données dans un Message et déclenche le Processus.
2. Le Processus fusionne le contenu du Message avec le modèle de formulaire et génère des données de sortie structurées.
3. Les données de sortie obtenues sont envoyées à un pilote et en post-traitement.
4. La sortie mise en forme est envoyée à sa destination.

Dans ce chapitre :

Configuration du Projet

Prérequis :

- Un Projet StreamServe ouvert dans Design Center.
- Un fichier d'entrée contenant des données d'entrée structurées en champs.

Pour créer le Message

1. Sélectionnez **Fichier > Nouveau > Message**. Un Message vide est créé.
2. Renommez le Message.

Pour créer l'Evénement

1. Cliquez avec le bouton droit de la souris dans la vue Message et sélectionnez **Ajouter un Evénement** et Type d'Evénement. Pour des données d'entrée structurées en champs, le type d'Evénement est StreamIN. Un nouvel Evénement est ajouté à la vue Message.
2. Renommez l'Evénement.
3. Cliquez avec le bouton droit de la souris sur cet Evénement et sélectionnez **Ouvrir**. L'outil Evénement s'ouvre.
4. Configurez l'Evénement. Voir la documentation sur *StreamServe Design Center*.
5. Enregistrez les données et quittez l'outil Evénement.

Pour créer le Processus

1. Cliquez avec le bouton droit de la souris dans la vue Message et sélectionnez **Ajouter un Processus > Adobe LiveCycle Designer ES2**. Un nouveau Processus est ajouté à la vue Message.
2. Renommez le Processus.
3. Cliquez avec le bouton droit de la souris sur ce Processus et sélectionnez **Ouvrir**. L'outil Processus s'ouvre et la boîte de dialogue Paramètres s'affiche. Voir *Boîte de dialogue Paramètres* à la page 84.
4. Dans la boîte de dialogue Paramètres, sélectionnez les options **Charger l'interface utilisateur de LiveCycle Designer au démarrage** et **Ajouter le Message sous la forme d'une connexion de données dans la Vue des données**.
5. Cliquez sur **OK**. La boîte de dialogue Paramètres se ferme et l'outil LiveCycle Designer se lance pour afficher une zone de dessin vide et le contenu de l'arborescence du Message en tant que connexion de données.
6. Configurez le Processus et déplacez les champs du Message vers la zone de dessin par "glisser-déplacer". Un objet de formulaire avec une liaison vers un champ est créé pour chaque champ.
7. Sélectionnez **Enregistrer**. Lorsque vous enregistrez pour la première fois, la boîte de dialogue Sélectionner une ressource pour le stockage du modèle XDP principal s'ouvre.

8 Sélectionnez la ressource et cliquez sur **OK**.

Pour finaliser le Projet

1 Configurez la Plate-forme.

2 Configurez le Runtime.

3 Exportez le Projet.

Voir la documentation sur *StreamServe Design Center*.

L'étape suivante consiste à déployer et à exécuter le Projet (voir *Déploiement et exécution du Projet* à la page 22).

Déploiement et exécution du Projet

Pour déployer le Projet

Vous déployez un Projet vers une application StreamServer dans Control Center :

1. Créez l'application StreamServer vers laquelle vous voulez déployer le Projet.
2. Déployez le Projet vers cette application.

Voir la documentation sur *StreamServe Control Center*.

Pour exécuter l'application StreamServer

Lancez l'application StreamServer dans Control Center en cliquant avec le bouton droit de la souris sur le noeud de l'application et sélectionnez **Démarrer**.

De même, vous arrêtez et redéployez des applications StreamServer dans Control Center. Voir la documentation sur *StreamServe Control Center*.

Utilisation de modèles

Note : Le chargement dynamique de modèles peut avoir un impact négatif sur les performances. Voir *Remarques relatives aux performances* à la page 12.

Références dans les modèles chargés dynamiquement

Les modèles comportant des dépendances (références externes à des fragments et des images) sont pris en charge si le Processus peut accéder aux fichiers référencés via :

- Des chemins statiques
- Des chemins relatifs par rapport à l'emplacement du modèle de formulaire dans un système de fichiers
- Un URL HTTP

Références dans les modèles du référentiel LiveCycle

Les modèles chargés dynamiquement à partir du référentiel LiveCycle ES2 ne peuvent pas contenir de références à d'autres ressources dans le référentiel. Cela signifie par exemple que les images doivent être incorporées ou utiliser des URL HTTP, des chemins de fichiers, des URI FTP, etc.

Versions d'application

Il peut exister plusieurs versions d'une ressource dans le référentiel LiveCycle ES2 qui sont liées à différentes versions de l'application LiveCycle. LiveCycle Production Print ES2 n'utilisera pas automatiquement la dernière version de l'application.

Dans ce chapitre :

Utilisation d'une variable StreamServe pour charger un modèle

Vous pouvez utiliser une variable StreamServe lorsqu'il est impossible de définir le chemin du modèle par le biais d'une expression SOM, ou lorsque le chemin doit être spécifié à l'aide d'un script dans StreamServe.

La variable est évaluée une fois pour chaque job, immédiatement après le lancement du Processus. Le modèle est donc utilisé pour tous les enregistrements traités dans le job.

La variable est une chaîne et les chemins de fichier et les URI (fichier, HTTP et URI de référentiel) sont pris en charge. Ils peuvent être absolus ou relatifs (par rapport au répertoire de travail StreamServer).

Pour les URI HTTP, vous pouvez utiliser l'authentification HTTP simple.

Exemple 1 *Exemples de chemins de fichier et d'URI*

- Chemin de fichier :
  `C:\templates\mytemplate.xdp`
  ou
  `../../MyTemplate.xdp`
- URI de fichier :
  `file:///D:/my%20templates/mytemplate.xdp`
- URI HTTP :
  `HTTP:// www.exampletemplate.com`
- URI de référentiel :
  `LC://myhost:8080/myfolder/myresource.xdp`

Exemple 2 *Sélection d'un modèle sur la base de données externes*

Le fichier du modèle utilisé pour une catégorie d'utilisateur particulière est stocké sur disque (ou dans une base de données).

Un script StreamServe Avant Processus est utilisé pour lire une valeur dans le fichier, selon la catégorie d'utilisateur, et permet d'attribuer le chemin du modèle à la variable. Cette dernière est alors reprise pour sélectionner le modèle dans le Processus.

Exemple de chemin absolu :

```
$template =”C:\templates\dynamic_invoice.xdp”;
```

Exemple de chemin relatif (à partir du répertoire de travail) :

```
$template = “../data/XDP_template/ dynamic_invoice.xdp”;
```

Exemple 3 *Sélection d'un modèle sur la base des méta-données*

Le chemin du modèle est envoyé à StreamServe sous forme de valeur d'en-tête HTTP.

Un script extrait cette valeur de l'en-tête HTTP et l'attribue à une variable. Cette dernière est alors reprise pour sélectionner le modèle dans le Processus.

Pour charger un modèle lors de l'exécution à l'aide d'une variable StreamServe

1. Dans l'outil Processus, sélectionnez **Fichier > Ouvrir un modèle**. La boîte de dialogue Sélection du modèle s'ouvre (voir *Boîte de dialogue Sélection du modèle*).
2. Sélectionnez **Variable StreamServe** et entrez le nom de la variable qui pointe vers le modèle.
3. En cas de connexion à un référentiel LiveCycle, entrez les informations de **connexion au référentiel Runtime**.
4. Si vous utilisez un URI HTTP et l'authentification HTTP simple, sélectionnez **Utiliser l'authentification HTTP simple** et entrez les identifiants de connexion.
5. Cliquez sur **OK**. La fenêtre de démarrage de l'outil Processus s'ouvre.

Utilisation d'une expression SOM pour charger un modèle

Utilisation de l'expression SOM

L'expression SOM s'avère utile lorsque le chemin d'accès à un modèle est accessible à partir du DOM de données et lorsque vous devez utiliser un modèle différent pour chacun des enregistrements d'un job d'arrière-plan.

L'expression SOM peut être évaluée une fois pour chaque enregistrement ou page traité. Il est donc possible d'utiliser des modèles différents pour chaque enregistrement ou page traité.

Les chemins de fichier et les URI sont pris en charge de la même façon que pour la variable StreamServe (voir *Utilisation d'une variable StreamServe pour charger un modèle* à la page 24).

Exemple 4 *Expression SOM*

Le chemin du modèle est enregistré dans un élément du fichier d'entrée.

Une expression SOM pointant vers cet élément permet de sélectionner le modèle dans le Processus.

Pour charger un modèle lors de l'exécution à l'aide d'une expression SOM

1. Dans l'outil Processus, sélectionnez **Fichier > Ouvrir un modèle**. La boîte de dialogue Sélection du modèle s'ouvre (voir *Boîte de dialogue Sélection du modèle*).
2. Sélectionnez **Expression SOM** et entrez l'expression SOM dans le DOM de données qui pointe vers un modèle.
3. En cas de connexion à un référentiel LiveCycle, entrez les informations de **connexion au référentiel Runtime**.
4. Si vous utilisez un URI HTTP et l'authentification HTTP simple, sélectionnez **Utiliser l'authentification HTTP simple** et entrez les identifiants de connexion.
5. Cliquez sur **OK**. La fenêtre de démarrage de l'outil Processus s'ouvre.

Chargement direct à partir du référentiel LiveCycle

Les modèles chargés à partir du référentiel LiveCycle le sont de façon statique lors de la conception. StreamServer charge les modèles à partir du référentiel au moment de l'exécution. L'utilisateur peut donc mettre à jour un modèle et stocker une nouvelle version dans le référentiel lors de l'exécution. Le serveur chargera le dernier modèle stocké.

Pour charger un modèle à partir du référentiel LiveCycle

1. Dans l'outil Processus, sélectionnez **Fichier > Ouvrir un modèle**. La boîte de dialogue Sélection du modèle s'ouvre (voir *Boîte de dialogue Sélection du modèle*).
2. Entrez les informations de **connexion au référentiel Runtime** pour le référentiel LiveCycle.
3. Sélectionnez l'option **A partir du référentiel LiveCycle** et accédez à un modèle du référentiel. Les informations de connexion au référentiel Runtime servent à se connecter au référentiel.
4. Cliquez sur **OK**. La fenêtre de démarrage de l'outil Processus s'ouvre.

Chargement de polices

Pour des raisons de performance, StreamServe charge toutes les polices au démarrage de StreamServer.

Lorsque vous utilisez des modèles statiques, le package d'exportation Design Center contient toutes les polices référencées.

Lorsque vous utilisez des modèles dynamiques, vous devez inclure manuellement les polices à utiliser.

Vous devez donc ajouter les polices au jeu de ressources d'un Projet.

Pour ajouter les polices au jeu de ressources d'un Projet

Importez manuellement les polices dans le Projet, comme l'explique la *documentation sur Design Center*.

Astuce - Variante

Vous pouvez également suivre la procédure suivante pour ajouter manuellement les polices : importez un ou plusieurs modèles XDP contenant toutes les polices que vous souhaitez utiliser en tant que ressources et associez-les une à une à l'outil Processus (LiveCycle).

Enregistrez le Processus après avoir attaché chaque ressource XPD. Toutes les polices utilisées seront alors automatiquement importées dans le Projet.

Vous pouvez, le cas échéant, supprimer les ressources XPD une fois l'opération terminée. Voir la documentation sur *Design Center*.

Utilisation de modèles externes

Vous pouvez utiliser un fichier de schéma XML, exporté d'un Message StreamServe, pour modifier et concevoir un modèle dans l'application autonome LiveCycle Designer ES. Le fichier exporté utilise le Message StreamServe comme connexion de données dans la vue Données. Le modèle peut être réimporté dans la solution StreamServe. Cela peut s'avérer utile, par exemple, si vous externalisez le développement et la gestion des modèles.

Vous pouvez créer un aperçu du schéma XML à l'aide de données exemples. Vous saisissez ces données pour chaque champ dans l'outil Evénement StreamServe, par exemple PageIN ou XMLIN.

Cela est particulièrement utile lorsque les données d'entrée n'ont pas un format de données XML (texte ASCII par exemple). LiveCycle Production Print ES2 transforme en interne l'entrée au format XML mais si vous voulez générer un aperçu dans LiveCycle Designer ES2, vous ne disposez généralement d'aucun fichier XML pour effectuer cette tâche. Vous pouvez exécuter la fonction Exporter le fichier XML d'aperçu pour en créer un dans ce scénario.

Exemple d'utilisation

Vous pouvez avoir externalisé la conception du modèle dans un concepteur externe. Vous créez deux fichiers exportés, un fichier de schéma de message et un fichier d'aperçu avec des données exemples, et les transmettez au concepteur externe. Lorsque le concepteur externe a terminé, vous réimportez le modèle dans Design Center.

Pour exporter un Message StreamServe

1. Dans l'outil Processus, sélectionnez **Fichier > Exporter le schéma de Message**. La boîte de dialogue Enregistrer sous s'ouvre.
2. Naviguez jusqu'à l'emplacement et le nom du fichier de schéma (XSD) et sélectionnez-les.

Pour créer un fichier XML d'aperçu avec les données exemples

1. Ouvrez la boîte de dialogue Paramètres et vérifiez que l'option **Ajouter le Message sous la forme d'une connexion de données dans la Vue des données** est activée.
2. Créez un fichier XML d'aperçu avec les données d'exemples, sélectionnez **Fichier > Exporter le fichier XML d'aperçu**.
3. Naviguez jusqu'à l'emplacement et le nom du fichier de schéma (XSD) et sélectionnez-les.

Pour réimporter un modèle exporté dans un jeu de ressources Design Center

1. Ouvrez la vue du jeu de ressources.
2. Sélectionnez la commande **Importer** et naviguez jusqu'au fichier du modèle à importer pour le sélectionner. La boîte de dialogue Paramètres du type de ressource s'ouvre.
3. Spécifiez le type de ressource **Modèle XDP** dans la liste déroulante et cliquez sur **OK**. La ressource est ajoutée au jeu de ressources.

Intégration LiveCycle Production Print ES2 et LiveCycle ES2

Accès de StreamServe Design Center au référentiel LiveCycle ES2

Il est possible de connecter StreamServe Design Center à un référentiel LiveCycle ES2, ce qui permet d'utiliser des modèles et d'autres ressources associées avec LiveCycle Production Print ES2 sans avoir à les importer via le système de fichiers. Les modèles et les fichiers associés sont importés du référentiel LiveCycle ES2 vers un jeu de ressources dans le Projet Design Center.

Appel des applications LiveCycle Production Print ES2 par LiveCycle ES2

LiveCycle ES2 peut appeler des applications StreamServer qui sont exposées via des services Web. Ces services Web peuvent être utilisés pour intégrer des applications StreamServer à des processus LiveCycle ES2 lors du traitement de documents.

Appel des processus LiveCycle par LiveCycle Production Print ES2

StreamServer peut appeler des processus LiveCycle qui sont déployés dans LiveCycle ES2 et exposés par le biais de services Web. Ces services Web peuvent être utilisés pour intégrer les processus LiveCycle dans le pipeline StreamServer lors du traitement de documents.

Dans ce chapitre :

Accès au référentiel Adobe LiveCycle ES2

Il est possible d'importer des modèles et les fichiers associés du référentiel LiveCycle ES2 vers Design Center et de les stocker en tant que jeux de ressources.

Lorsque vous accédez au référentiel LiveCycle ES2 à partir de Design Center, vous pouvez effectuer les opérations suivantes :

- Naviguer dans le référentiel LiveCycle. Le navigateur affiche les informations suivantes :
  - Si la ressource existe déjà en local dans le jeu de ressources Design Center.
  - Si elle a été mise à jour dans le référentiel et si elle doit l'être en local dans le jeu de ressources Design Center.
  - Si elle a été mise à jour en local dans le jeu de ressources Design Center.
  - Si elle a été retirée du référentiel.

  Voir *Icônes utilisées dans le référentiel LiveCycle ES2* à la page 97.
- Importer des ressources et leurs dépendances à partir du référentiel LiveCycle ES2.
- Mettre à jour des ressources déjà importées à partir du référentiel LiveCycle ES2.

Vous ne pouvez modifier aucune donnée dans le référentiel LiveCycle ES2 lorsque vous y accédez à partir de Design Center. Vous ne disposez que d'un accès en lecture seule.

Note : Vous pouvez définir des connexions à plusieurs référentiels, mais vous ne pouvez créer et mettre à jour des ressources qu'à partir d'un seul référentiel à la fois.

Connexion aux référentiels LiveCycle

Vous pouvez définir et activer les connexions à un référentiel LiveCycle ES2 à partir de Design Center. Vous ne pouvez activer qu'une seule connexion à la fois. Celle-ci ne sera active que pendant la communication (pendant l'importation et la mise à jour).

Sélectionnez **Outils > Sélectionner une connexion au référentiel Adobe LiveCycle**. La boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle s'ouvre (voir *Boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle*).

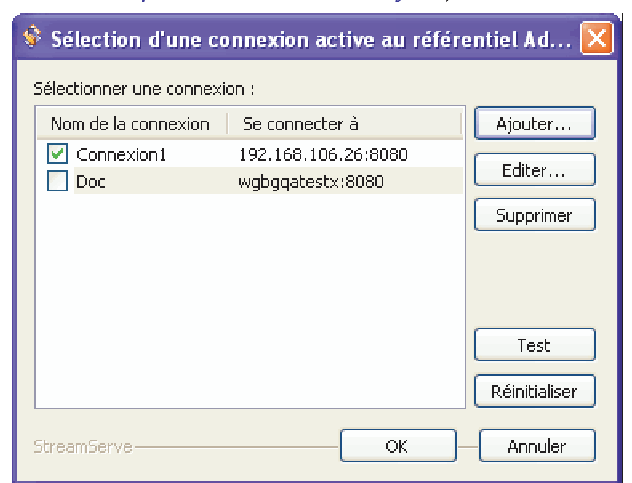


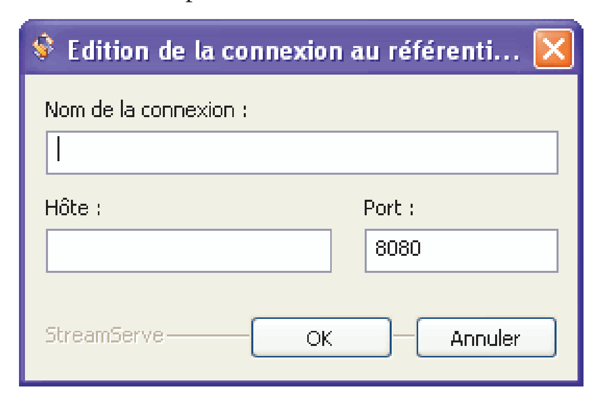
Figure 1 Boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle

Pour activer une connexion

Cochez la case requise pour activer la connexion. La connexion sélectionnée sera activée lorsque ce sera nécessaire (c'est-à-dire pendant l'importation et la mise à jour à partir du référentiel).

Pour ajouter/modifier une connexion

1. Cliquez sur **Ajouter** ou **Editer**. La boîte de dialogue Edition de la connexion au référentiel Adobe LiveCycle s'ouvre.
2. Définissez les paramètres.



- **Nom de la connexion** – Choisissez un nom approprié pour la connexion.
- **Hôte** – Nom d'hôte ou adresse IP du serveur sur lequel le référentiel est installé.
- **Port** – Port utilisé pour communiquer avec l'hôte.

3 Cliquez sur **OK**. La connexion doit exister et être disponible ; lorsque la connexion est ajoutée, elle est établie et identifiée. Si elle n'existe pas, un message d'erreur s'affiche.

Pour supprimer une connexion

1 Sélectionnez la connexion (en surbrillance) à supprimer.

2 Cliquez sur **Supprimer**.

Pour tester une connexion

Vous pouvez tester une connexion à un référentiel Adobe LiveCycle pour vérifier si elle fonctionne.

1 Sélectionnez la connexion (en surbrillance) à tester.

2 Cliquez sur **Test**. Vous êtes invité à saisir vos identifiants de connexion.

3 Entrez vos identifiants et cliquez sur **OK**.

Importation d'une ressource à partir du référentiel Adobe LiveCycle ES2

Lorsque vous importez une ressource du référentiel LiveCycle ES2, le système crée une copie locale de la ressource dans le jeu de ressources Design Center.

La ressource est ajoutée dans une structure de chemins et de fichiers qui reflète la structure dans le référentiel LiveCycle ES2.

Note : Ne modifiez pas la structure dans le jeu de ressources Design Center. Les références aux fragments et aux images peuvent être endommagées en cas de modification des positions relatives internes.

Vous pouvez choisir d'importer une ressource avec ou sans dépendances.

Pour importer une ressource

1 Sélectionnez **Ressources > Importer du référentiel Adobe LiveCycle**. La boîte de dialogue Sélectionner une ressource s'ouvre.

2 Accédez à la ressource à importez et double-cliquez dessus. La boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle s'ouvre (voir *Boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle*).

   L'option **Toujours extraire la version principale de toutes les ressources** est activée par défaut. Cela signifie que la dernière version de la ressource et ses dépendances seront importées.

3 Si vous voulez importer une autre version que la version principale, décochez la case et sélectionnez une version dans la liste déroulante.

4 Sélectionnez la ressource et les dépendances à importer et cliquez sur **OK**.

Mise à jour des ressources à partir du référentiel Adobe LiveCycle ES2

Pour mettre à jour une ressource et ses dépendances

Si la ressource d'origine (dans le référentiel LiveCycle ES2) a été modifiée, vous pouvez mettre à jour la copie locale dans le jeu de ressources Design Center.

1. Cliquez avec le bouton droit de la souris sur la ressource dans la vue Jeu de ressources et sélectionnez **Mettre à jour selon la source**. La boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle s'ouvre (voir *Boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle*) ; elle affiche l'état de la ressource locale et de ses dépendances par rapport à la ressource du référentiel. Voir *Icônes utilisées dans le référentiel LiveCycle ES2* à la page 97.
2. Sélectionnez la ressource et les dépendances à importer.
3. Cliquez sur **OK**. La ressource et les dépendances sélectionnées seront enregistrées dans la même structure de chemins et de fichiers que dans le référentiel Adobe LiveCycle.

Pour mettre à jour plusieurs ressources

Lorsqu'un grand nombre de ressources ont été modifiées dans le référentiel LiveCycle ES2, vous pouvez mettre à jour plusieurs ressources en une seule fois.

1. Sélectionnez **Ressources > Mettre à jour toutes les ressources Adobe LiveCycle**. La boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle s'ouvre (voir *Boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle*) et affiche la ressource importée ainsi que son état par rapport aux ressources du référentiel. Voir *Icônes utilisées dans le référentiel LiveCycle ES2* à la page 97.

Note : Les dépendances ne sont pas mises à jour lorsque vous utilisez la commande **Mettre à jour toutes les ressources Adobe LiveCycle**.

2. Sélectionnez les ressources à mettre à jour en cochant les cases correspondantes ou en cochant **Sélectionner tout.**
3. Cliquez sur **OK**. Les ressources sélectionnées seront enregistrées dans la même structure de chemins et de fichiers que dans le référentiel LiveCycle ES2.

Appel de LiveCycle Production Print ES2 à partir de LiveCycle ES2

LiveCycle ES2 peut appeler des applications StreamServer qui sont exposées via des services Web. Ces services Web peuvent être utilisés pour intégrer des applications StreamServer à des processus LiveCycle ES2 lors du traitement de documents.

Données d'entrée et de sortie

Le service StreamServer est entièrement générique. Il est possible d'envoyer n'importe quel type de données à StreamServer, par exemple, un fichier de données XML à fusionner avec un modèle de formulaire.

Le résultat obtenu par StreamServer est également générique. Il peut s'agir aussi bien d'un fichier d'impression que d'un message d'état, selon la configuration StreamServer.

Composant DSC (Document Service Component) utilisé pour l'intégration

Le composant DSC (Document Service Component) appelé LiveCycle Production Print DSC est développé dans ce but. Il peut être utilisé dans n'importe quel processus LiveCycle ES2 pour le transfert de données vers et depuis StreamServer.

LiveCycle Production Print DSC est fourni sous la forme d'un fichier .jar (`lcppdsc.jar`) sur le support d'installation. Il peut être déployé dans LiveCycle ES2 par le biais de l'outil Workbench.

Connecteurs StreamServer utilisés pour l'intégration

Les services Web sont exposés par StreamServer via des connecteurs d'entrée Service Request. StreamServer reçoit le job de LiveCycle ES2 via le connecteur Service Request et peut renvoyer les données de sortie traitées à LiveCycle ES2 via n'importe quel connecteur de sortie.

Passerelle de services

Une passerelle de services doit être active. Elle gère les appels de service Web entre LiveCycle ES2 et StreamServer.

Projet exemple

Un Projet Design Center, `sampleproject.dcpackage`, est fourni sur le support d'installation. Ce Projet donne un exemple d'utilisation de ce type : il comprend le processus LiveCycle ES2 ainsi que LiveCycle Production Print. Il est recommandé d'étudier et de tester ce Projet exemple. Nous vous conseillons également de prendre modèle sur ce Projet exemple pour développer vos propres processus intégrés. Voir *Projet exemple* à la page 57 pour plus d'informations sur le Projet exemple.

Caractéristiques de LiveCycle Production Print DSC

Le service LiveCycle Production Print ES2 Workbench a trois fonctions :

- *Post* – Envoi d'un job de LiveCycle ES2 à StreamServer. Aucune information d'état n'est renvoyée.
- *Run* – Envoi d'un job de LiveCycle ES2 à StreamServer. Des informations d'état sont renvoyées lorsque le job de sortie de StreamServer est terminé.
- *Generate* – Envoi d'un job de LiveCycle ES2 à StreamServer pour traitement et réception du job traité et des informations d'état sous forme de réponse émanant de StreamServer.

Post

Utilisez cette fonction si vous voulez seulement que LiveCycle ES2 envoie un job à StreamServer pour traitement ultérieur et dans le cas où aucune information d'état n'est requise une fois le job de sortie de StreamServer terminé.

Paramètres

Les paramètres ci-dessous s'appliquent à cette fonction.

Paramètre	Type et sous-type	Description
Remote Endpoint	**Type** : chaîne	Adresse de la passerelle de services. Par exemple : `http://localhost:2718`
Remote Service Name	**Type** : chaîne	Nom (sensible à la casse) du service StreamServer à appeler. Il doit être absolument identique au `Nom de service` du connecteur d'entrée Service Request utilisé par StreamServer pour récupérer le job. Voir *Création d'un connecteur d'entrée Service Request* à la page 42.
Connection Timeout	**Type** : nombre entier	Délai maximum (en secondes) d'attente de StreamServer pour la récupération du job. Un délai <=0 signifie pas de temporisation.
Input Data	**Type** : document	Job (document, cycle de traitement par lot, etc.) et type de contenu du job que StreamServer doit traiter.
Template	**Type** : document	Modèle XDP que StreamServer doit utiliser lorsqu'il traite le job.
Additional Input Parameters	**Type** : liste **Sous-type** : paramètre d'entrée	Variables à communiquer à StreamServer. StreamServer doit utiliser la fonction de script `GetConnectorValue` pour accéder à ces variables.

Run

Utilisez cette fonction si vous voulez seulement que LiveCycle ES2 envoie un job à StreamServer pour traitement ultérieur et si vous voulez que StreamServer renvoie les informations d'état une fois le job de sortie terminé.

Paramètres

Les paramètres ci-dessous s'appliquent à cette fonction.

Paramètre	Type et sous-type	Description
Remote Endpoint	**Type** : chaîne	Adresse de la passerelle de services. Par exemple : `http://localhost:2718`
Remote Service Name	**Type** : chaîne	Nom (sensible à la casse) du service StreamServer à appeler. Il doit être absolument identique au `Nom de service` du connecteur d'entrée Service Request utilisé par StreamServer pour récupérer le job. Voir *Création d'un connecteur d'entrée Service Request* à la page 42.
Connection Timeout	**Type** : nombre entier	Délai maximum (en secondes) d'attente d'une réponse de StreamServer. Si le délai de temporisation est écoulé lorsque StreamServer traite le job, ce dernier n'est pas supprimé de la base de données File d'attente. Un délai <=0 signifie pas de temporisation.
Input Data	**Type** : document	Job (document, cycle de traitement par lot, etc.) et type de contenu du job que StreamServer doit traiter.
Template	**Type** : document	Modèle XDP que StreamServer doit utiliser lorsqu'il traite le job.
Additional Input Parameters	**Type** : liste **Sous-type** : paramètre d'entrée	Variables à communiquer à StreamServer. StreamServer doit utiliser la fonction de script `GetConnectorValue` pour accéder à ces variables.
Returned Status and Documents	**Type** : état du résultat et document(s)	Etat et documents renvoyés par StreamServer.
Paramètres secondaires relatifs à l'état et aux documents renvoyés		
statusCode	**Type** : nombre entier	Code d'état renvoyé par StreamServer. 0 : OK 1 : Avertissement

Paramètre	Type et sous-type	Description
statusMessage	**Type** : chaîne	Informations complémentaires relatives à l'état. Contient des informations détaillées sur le code d'état renvoyé par StreamServer.

Generate

Utilisez cette fonction si vous voulez que LiveCycle ES2 envoie un job à StreamServer pour traitement et récupère ensuite le job traité sous forme de réponse de StreamServer.

Paramètres

Les paramètres ci-dessous s'appliquent à cette fonction.

Paramètre	Type et sous-type	Description
Remote Endpoint	**Type** : chaîne	Adresse de la passerelle de services. Par exemple : `http://localhost:2718`
Remote Service Name	**Type** : chaîne	Nom (sensible à la casse) du service StreamServer à appeler. Il doit être absolument identique au `Nom de service` du connecteur d'entrée Service Request utilisé par StreamServer pour récupérer le job. Voir *Création d'un connecteur d'entrée Service Request* à la page 42.
Connection Timeout	**Type** : nombre entier	Délai maximum (en secondes) d'attente d'une réponse de StreamServer. Si le délai de temporisation est écoulé lorsque StreamServer traite le job, ce dernier n'est pas supprimé de la base de données File d'attente. Un délai <=0 signifie pas de temporisation.
Input Data	**Type** : document	Job (document, cycle de traitement par lot, etc.) et type de contenu du job que StreamServer doit traiter.
Template	**Type** : document	Modèle XDP que StreamServer doit utiliser lorsqu'il traite le job.
Additional Input Parameters	**Type** : liste **Sous-type** : paramètre d'entrée	Variables à communiquer à StreamServer. StreamServer doit utiliser la fonction de script `GetConnectorValue` pour accéder à ces variables.
Returned Status and Documents	**Type** : état du résultat et document(s)	Etat et documents renvoyés par StreamServer.

Paramètre	Type et sous-type	Description
Paramètres secondaires relatifs à l'état et aux documents renvoyés		
statusCode	**Type** : nombre entier	Code d'état renvoyé par StreamServer. 0 : OK 1 : Avertissement
statusMessage	**Type** : chaîne	Informations complémentaires relatives à l'état. Contient des informations détaillées sur le code d'état renvoyé par StreamServer.
documents	**Type** : liste **Sous-type** : document	Job, et type de contenu du job, renvoyé par StreamServer.

Gestion des erreurs

En cas d'erreur, l'exception `ProductionPrintException` est déclenchée. Les codes d'erreur sont décrits dans le tableau ci-dessous.

Code	Description
`-1`	Erreur du serveur de service. Il s'agit d'une erreur qui ne peut pas être corrigée sur le client simplement en effectuant une nouvelle tentative. Par exemple : mémoire ou espace disque insuffisant sur le serveur.
`-2`	Erreur du client. Il s'agit d'une erreur qui peut être corrigée sur le client. Par exemple : nom de service non valable ou temporisation trop courte.
`-3`	Erreur d'appel. Une enveloppe SOAP non valable a été envoyée au service Web.
`-4`	Erreur d'un périphérique distant. Par exemple : panne de réseau ou terminal spécifié non valable.
`-5`	Erreur de données de sortie. Le client n'a pas pu recevoir les données de sortie du serveur.
`-6`	Erreur d'adressage. Un terminal non valable a été spécifié pour le service Web.
`-7`	Erreur d'entrée/sortie locale. Erreur d'entrée/sortie du composant DSC. Par exemple : espace disque insuffisant sur l'hôte DSC.

Code	Description
-8	Erreur DSC générique. Le message d'erreur de l'exception contient de plus amples détails.

Configuration StreamServer

La configuration StreamServer inclut un connecteur d'entrée Service Request et l'Evénement, le Processus, le connecteur de sortie et les files d'attente appropriés.

Connecteur d'entrée Service Request

Ce connecteur expose le service Web dans LiveCycle ES2 et y récupère le job.

Evénement et Processus

L'Evénement et les Processus sont configurés selon les procédures Design Center standard.

Connecteur de sortie

Dans un scénario dans lequel StreamServer génère la sortie finale, le connecteur de sortie est configuré selon les procédures Design Center standard.

Dans un scénario dans lequel LiveCycle ES2 génère la sortie finale, le connecteur de sortie doit également être configuré pour renvoyer le job dans la réponse du service Web.

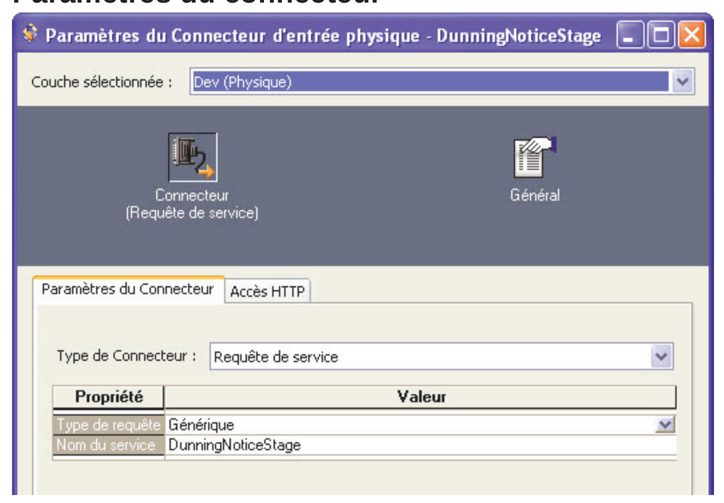
Files d'attente

Les connecteurs d'entrée et de sortie doivent être connectés à des files d'attente.

Création d'un connecteur d'entrée Service Request

Vous créez un connecteur d'entrée Service Request de la même façon que vous créez d'autres connecteurs d'entrée dans Design Center.

Paramètres du connecteur



Paramètre	Description
Type de requête	Sélectionnez **Générique**.
Nom du service	Nom du service Web à exposer dans LiveCycle ES2.

Activation d'une réponse de service

Dans un scénario dans lequel LiveCycle ES2 génère la sortie finale, StreamServer doit être configuré pour renvoyer sa sortie à LiveCycle ES2 dans la réponse du service Web. Cette opération s'effectue dans la configuration du connecteur de sortie. Il est possible d'utiliser n'importe quel type de connecteur de sortie, par exemple un connecteur Null.

Note : Le connecteur de sortie doit être connecté à une file d'attente de sortie.

Pour activer la réponse de service

1 Dans Design Center, activez la couche de Plate-forme générique.

2 Double-cliquez sur le connecteur de sortie. La boîte de dialogue Paramètres de connecteur de sortie s'ouvre.

3 Cliquez sur l'icône **Général**, sélectionnez **Inclure le résultat dans la réponse de service**, puis cliquez sur **OK**.

Récupération des variables

S'il existe des variables dans la requête de service de LiveCycle ES2, StreamServer doit utiliser la fonction de script `GetConnectorValue` pour récupérer ces variables. Voir le manuel StreamServe *Scripting reference* pour plus d'informations sur cette fonction de script.

Appel de processus LiveCycle ES2 à partir de LiveCycle Production Print ES2

StreamServer peut appeler des processus LiveCycle qui sont déployés dans LiveCycle ES2 et exposés par le biais de services Web. Ces services Web peuvent être utilisés pour intégrer les processus LiveCycle dans le pipeline StreamServer lors du traitement de documents.

Vous pouvez appeler les processus créés et activés via LiveCycle Workbench ES2 en envoyant la requête d'appel appropriée (requête SOAP) à LiveCycle ES2.

Filtre LiveCycle et connecteur de sortie LiveCycle

Il existe deux façons d'appeler des requêtes de StreamServer vers LiveCycle ES2 :

- Connecteur de sortie LiveCycle – utilisé lorsque LiveCycle ES2 génère la sortie finale. Voir *Connecteur de sortie LiveCycle* à la page 44.
- Filtre LiveCycle – utilisé lorsque StreamServer génère la sortie finale. Voir *Filtre LiveCycle* à la page 49.

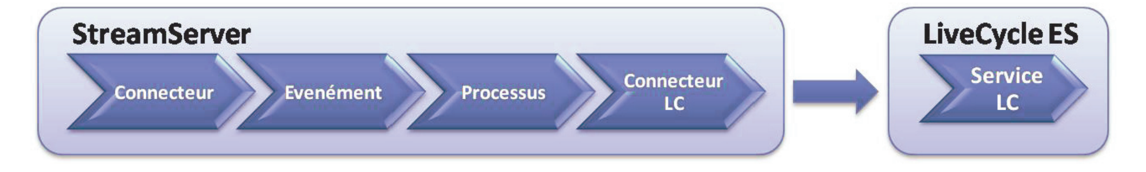
Projet exemple

Un Projet Design Center, `sampleproject.dcpackage`, est fourni sur le support d'installation. Ce Projet donne un exemple d'utilisation de ce type : il comprend le processus LiveCycle ES2 ainsi que LiveCycle Production Print. Il est recommandé d'étudier et de tester ce Projet exemple. Nous vous conseillons également de prendre modèle sur ce Projet exemple pour développer vos propres processus intégrés. Voir *Projet exemple* à la page 57 pour plus d'informations sur le Projet exemple.

Connecteur de sortie LiveCycle

Le connecteur de sortie LiveCycle est utilisé lorsque LiveCycle ES2 génère la sortie finale.

Exemple – Utilisation du connecteur de sortie LiveCycle



1. StreamServer reçoit les données d'entrée via un connecteur d'entrée.
2. StreamServer utilise la configuration Evénement/Processus appropriée pour créer des documents.
3. Le connecteur de sortie LiveCycle appelle le processus LiveCycle ES2 approprié et envoie les documents dans la requête.
4. Le processus LiveCycle ES2 traite les documents et génère la sortie finale.

Création d'un processus LiveCycle ES2 activé par un connecteur de sortie

Vous créez et activez le processus selon la procédure décrite dans la documentation de LiveCycle Workbench ES2. Pour activer le connecteur de sortie LiveCycle afin d'appeler le service déployé, vous devez ajouter les variables d'entrée suivantes au processus LiveCycle.

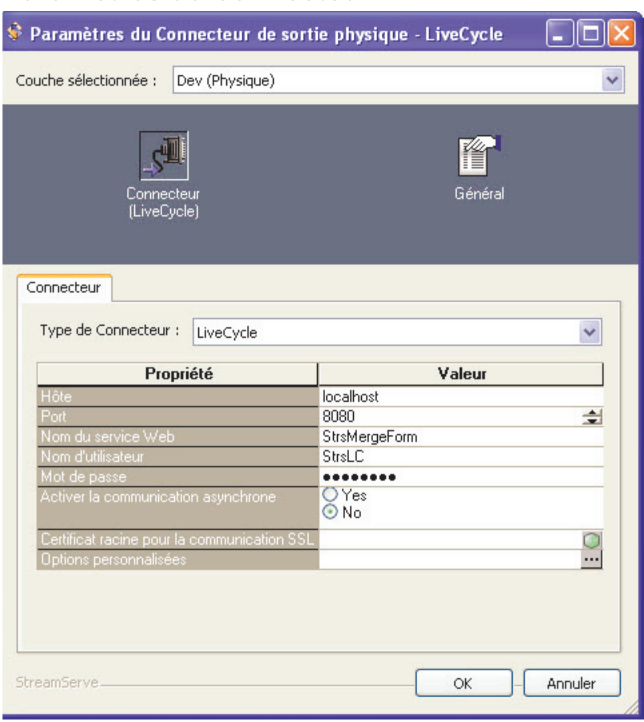
Nom de la variable	Type	Commentaire
`inputDoc`	`document`	Obligatoire
`optionsMap`	`map`	Facultatif. Est utilisée si des clés personnalisées sont spécifiées dans les paramètres du filtre LiveCycle.

Seuls les processus LiveCycle qui suivent cette interface peuvent être appelés par un connecteur de sortie LiveCycle.

Création d'un connecteur de sortie LiveCycle

Vous créez un connecteur de sortie LiveCycle de la même façon que vous créez d'autres connecteurs de sortie dans Design Center.

Paramètres du connecteur



Paramètre	Description
Hôte	Nom d'hôte ou adresse IP du serveur qui héberge LiveCycle ES2. Par exemple : `localhost`
Port	Port utilisé par le serveur LiveCycle ES2. Par exemple : `8080`
Nom du service Web	Nom (sensible à la casse) du service à appeler. Ce nom doit être identique au nom du processus correspondant créé dans LiveCycle Workbench ES2.
Nom d'utilisateur	Nom d'utilisateur permettant de se connecter au serveur hébergeant LiveCycle ES2. Ce paramètre est utilisé dans le cas de l'authentification HTTP de base.

Paramètre	Description
Mot de passe	Mot de passe permettant de se connecter au serveur hébergeant LiveCycle ES2. Ce paramètre est utilisé dans le cas de l'authentification HTTP de base.
Activer la communication asynchrone	**Oui** Effectue des appels asynchrones au service. Cette option est utilisée lorsque vous appelez des services LiveCycle sur une longue durée. **Non** Effectue des appels synchrones au service. Cette option est utilisée lorsque vous appelez des services LiveCycle sur une courte durée.
Intervalle d'interrogation asynchrone	Ce paramètre n'est utilisé qu'avec les appels asynchrones. Il s'agit de l'intervalle (en millisecondes) utilisé pour vérifier la réponse à la requête d'appel.
Certificat racine pour la communication SSL	Certificat racine employé lorsque le protocole HTTPS est utilisé comme protocole de service Web (communication sécurisée). Ce certificat doit être disponible à partir d'un jeu de ressources connecté à la Plate-forme.
Options personnalisées	Liste de clés personnalisées (paires valeurs/clés) à ajouter dans la requête d'appel. Pour pouvoir gérer les clés personnalisées, le service doit disposer d'une variable nommée `optionsMap` du type `map`. Toutes les clés personnalisées définies ici seront ajoutées à la variable `optionsMap` dans le service appelé. Les valeurs générées peuvent être extraites dans le processus LiveCycle destinataire à l'aide d'une expression XPath dans le processus LiveCycle. Les mots de passe requis pour créer des fichiers PDF protégés par mot de passe sont des exemples de clés personnalisées. Par exemple : **Clé :** `pdfpassword` **Valeur :** `encrypted`

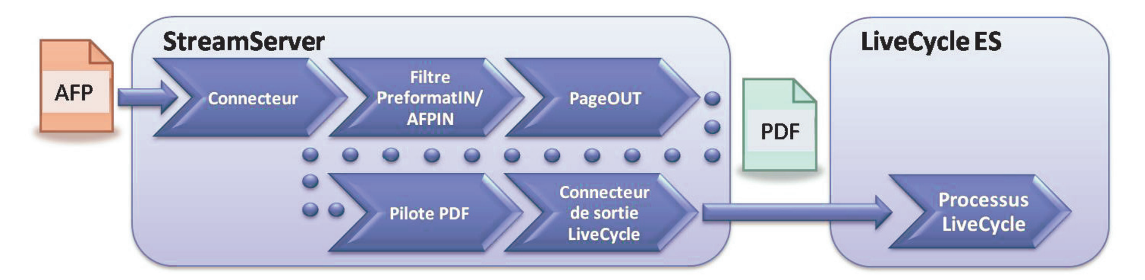
Exemple d'utilisation

Besoin

Un processus de gestion nécessite qu'un fichier AFP de factures soit converti au format PDF, puis envoyé à LiveCycle ES2 pour être enregistré dans LiveCycle Content Services ES2.

Actions

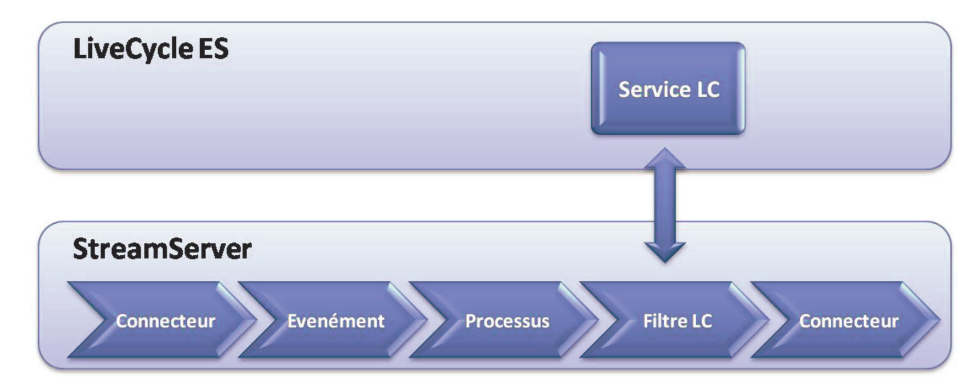
StreamServer est ajouté au pipeline. StreamServer récupère les données d'entrée AFP via un connecteur d'entrée, un filtre AFPIN et un Evénement PreformatIN. Les données AFP sont ensuite converties en données PDF via un Processus PageOUT et un pilote PDF. La sortie PDF est finalement envoyée à un processus LiveCycle via un connecteur de sortie LiveCycle.



Filtre LiveCycle

Le filtre LiveCycle est utilisé lorsque StreamServer génère la sortie finale.

Exemple – Utilisation du filtre LiveCycle



1. StreamServer reçoit les données d'entrée via un connecteur d'entrée.
2. StreamServer utilise la configuration Evénement/Processus appropriée pour créer des documents.
3. Le filtre LiveCycle appelle le processus LiveCycle ES2 approprié et envoie les documents dans la requête.
4. Le processus LiveCycle ES2 traite les documents et envoie les documents traités dans la réponse à StreamServer.
5. StreamServer génère la sortie finale via un connecteur de sortie.

Si le service Web tombe en panne avant que les documents ne soient envoyés dans la réponse de service Web, aucune sortie n'est générée. Dans ce cas, un message d'erreur est enregistré dans le journal.

Création d'un processus LiveCycle ES2 activé par un filtre

Vous créez et activez le processus selon la procédure décrite dans la documentation de LiveCycle Workbench ES2. Pour activer le filtre LiveCycle afin d'appeler le service déployé, vous devez ajouter les variables d'entrée et de sortie suivantes au processus LiveCycle.

Entrée/Sortie	Nom de la variable	Type	Commentaire
Entrée	`inputDoc`	`document`	Obligatoire
Entrée	`optionsMap`	`map`	Facultatif. Est utilisée si des clés personnalisées sont spécifiées dans les paramètres du filtre LiveCycle.
Sortie	`outputDoc`	`document`	Obligatoire

Seuls les processus LiveCycle qui suivent cette interface peuvent être appelés par un filtre LiveCycle.

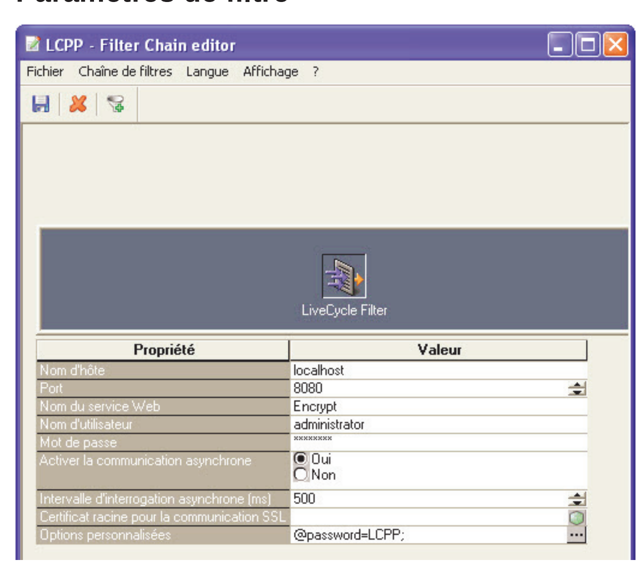
Création d'un filtre LiveCycle

Vous créez un filtre LiveCycle de la même façon que vous créez d'autres filtres de sortie dans Design Center. Vous devez pour cela créer une ressource de chaîne de filtres, créer et configurer le filtre LiveCycle dans la chaîne de filtres et connecter cette dernière au connecteur de sortie approprié.

Pour créer et appliquer un filtre LiveCycle

1. Créez une ressource de chaîne de filtres dans un jeu de ressources connecté à la Plate-forme.
2. Ajoutez un filtre LiveCycle à la chaîne de filtres.
3. Configurez le filtre (voir les paramètres de filtre ci-dessous) et enregistrez la ressource de chaîne de filtres.
4. Ajoutez la chaîne de filtres au connecteur de sortie approprié.

Paramètres de filtre



Paramètre	Description
Nom d'hôte	Nom d'hôte ou adresse IP du serveur qui héberge LiveCycle ES2. Par exemple : `localhost`
Port	Port utilisé par le serveur LiveCycle ES2. Par exemple : `8080`
Nom du service Web	Nom (sensible à la casse) du service à appeler. Ce nom doit être identique au nom du processus correspondant créé dans LiveCycle Workbench ES2.

Paramètre	Description
Nom d'utilisateur	Nom d'utilisateur permettant de se connecter au serveur hébergeant LiveCycle ES2. Ce paramètre est utilisé dans le cas de l'authentification HTTP de base.
Mot de passe	Mot de passe permettant de se connecter au serveur hébergeant LiveCycle ES2. Ce paramètre est utilisé dans le cas de l'authentification HTTP de base.
Activer la communication asynchrone	**Oui** Effectue des appels asynchrones au service. Cette option est utilisée lorsque vous appelez des services LiveCycle sur une longue durée. **Non** Effectue des appels synchrones au service. Cette option est utilisée lorsque vous appelez des services LiveCycle sur une courte durée.
Intervalle d'interrogation asynchrone	Ce paramètre n'est utilisé qu'avec les appels asynchrones. Il s'agit de l'intervalle (en millisecondes) utilisé pour vérifier la réponse à la requête d'appel.
Certificat racine pour la communication SSL	Certificat racine employé lorsque le protocole HTTPS est utilisé comme protocole de service Web (communication sécurisée). Ce certificat doit être disponible à partir d'un jeu de ressources connecté à la Plate-forme.

Paramètre	Description
Options personnalisées	Liste de clés personnalisées (paires valeurs/ clés) à ajouter dans la requête d'appel. Pour pouvoir gérer les clés personnalisées, le service doit disposer d'une variable nommée `optionsMap` du type `map`. Toutes les clés personnalisées définies ici seront ajoutées à la variable `optionsMap` dans le service appelé. Les valeurs générées peuvent être extraites dans le processus LiveCycle destinataire à l'aide d'une expression XPath dans le processus LiveCycle. Les mots de passe requis pour créer des fichiers PDF protégés par mot de passe sont des exemples de clés personnalisées. Par exemple : **Clé :** `pdfpassword` **Valeur :** `encrypted`

Exemple d'utilisation

Besoin

Un utilisateur StreamServer doit crypter des documents PDF, mais le cryptage ne peut pas être effectué à l'aide d'une fonctionnalité StreamServer. Vous pouvez dans ce cas appeler un processus LiveCycle pour crypter les documents PDF.

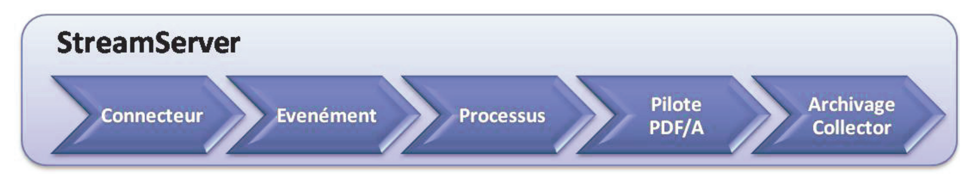


Figure 2 Avant le filtre LiveCycle

Actions

Un filtre LiveCycle est ajouté après le pilote PDF. Le filtre envoie les documents PDF formatés à un processus LiveCycle. Le processus crypte les documents et les renvoie à StreamServer via le filtre LiveCycle.

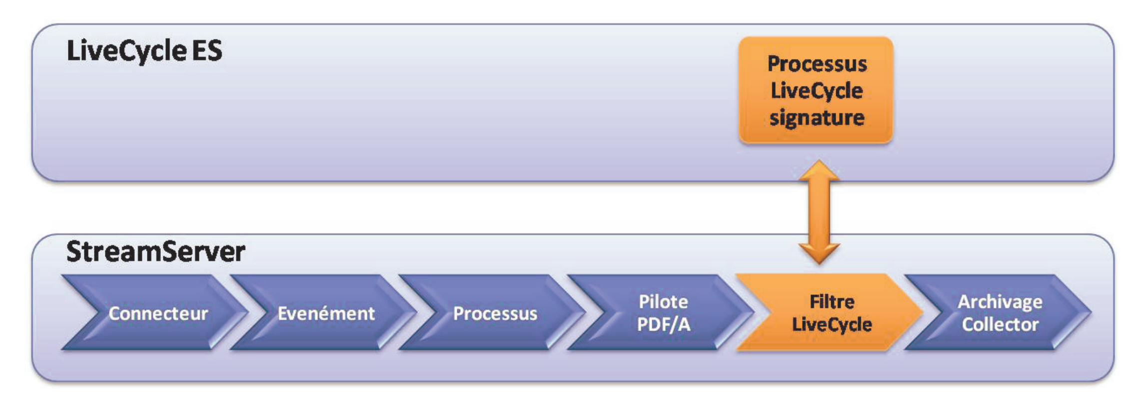


Figure 3 Après le filtre LiveCycle

Paramètres généraux du connecteur et du filtre LiveCycle

En plus des paramètres définis pour le filtre LiveCycle et le connecteur de sortie LiveCycle, vous pouvez avoir besoin de modifier certains paramètres généraux dans le fichier de configuration `strslcfilter.config.xml`. Ce fichier se trouve à l'emplacement suivant :

`<installation_StreamServe>\Services\XFA\1.3.0\Service`

```
<?xml version="1.0" encoding="utf-8"?>
<lcfilter>
  <setting key="maxinlinesize">65536</setting>
  <setting key="timeout">120</setting>
  <setting key="retries">5</setting>
 </lcfilter>
```

Figure 4 strslcfilter.config.xml – exemple.

Ces paramètres s'appliquent à tous les filtres et connecteurs de sortie LiveCycle.

Clé	Description
`maxinlinesize`	Taille maximale (en octets) autorisée d'un document pour pouvoir l'incorporer avec un codage base64. Si la taille maximale est dépassée, le document est alors enregistré comme attachement DIME.
`timeout`	Délai (en secondes) d'attente d'une réponse à la requête. Si ce délai est dépassé, la connexion est arrêtée.
`retries`	Nombre de tentatives en cas d'erreur de communication.

Projet exemple

Le Projet exemple peut être utilisé dans les cas suivants :

- Dans un but éducatif
- Pour vérifier que LiveCycle Production Print ES2 a été correctement installé et configuré

Le Projet exemple illustre également l'intégration des processus LiveCycle ES2 et LiveCycle Production Print ES2. Les exemples d'intégration fournis peuvent servir de modèles lors de la mise en oeuvre.

Le Projet exemple `SampleProject.dcpackage` est disponible dans le dossier `Extras\sampleproject` sur le support d'installation. Il est également installé par défaut dans le répertoire

`C:<répertoire installation StreamServe>\Services\XFA\1.3.0\Tool`

Prérequis

Pour exécuter le Projet exemple, les Processus et ressources du Projet doivent être installés.

Si vous utilisez un serveur LiveCycle (c'est-à-dire si vous appelez des applications StreamServer à partir de LiveCycle ES)2, le composant DSC doit être installé.

Voir le *manuel d'installation de LiveCycle Production Print ES2*.

Dans ce chapitre :

Exécution du Projet exemple

Vous devez décompresser le Projet exemple avant de pouvoir l'exporter et le déployer.

Pour décompresser le fichier du Projet exemple

1. Ouvrez Design Center.
2. Sélectionnez **Fichier > Décompresser le Projet**.
3. Recherchez et ouvrez le fichier de package. La boîte de dialogue Décompression du projet s'ouvre.
4. Spécifiez l'emplacement où vous souhaitez décompresser les fichiers du projet et cliquez sur **OK.**

Pour exporter et déployer le Projet exemple

Dans Design Center, exportez le Projet. Voir la documentation sur *StreamServe Design Center*.

Pour déployer le Projet exemple

Dans Control Center :

1. Créez l'application StreamServer vers laquelle vous voulez déployer le Projet exemple.
2. Déployez le Projet exemple vers l'application StreamServer.

Voir la documentation sur *StreamServe Control Center*.

Pour démarrer l'application StreamServer

Lancez l'application StreamServer dans Control Center en cliquant avec le bouton droit de la souris sur le noeud de l'application StreamServer et sélectionnez **Démarrer**.

De même, vous arrêtez et redéployez des applications StreamServer à partir de Control Center. Voir la documentation sur *StreamServe Control Center*.

Vérification de l'installation de base

Pour vérifier les fonctionnalités de base de l'installation

1. Décompressez, exportez et déployez le Projet. Suivez les instructions de la section *Exécution du Projet exemple* à la page 58.
2. Créez un dossier `input` dans le répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\Dev`
3. Copiez le fichier `Purchase Order.xml` du répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\data\samples` dans le dossier `input`.
4. Lisez le fichier journal dans Control Center pour vérifier que le fichier a été traité.
5. Vérifiez que le fichier `purchaseorder.pdf` est copié dans le répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\Dev\output`

Pour vérifier le Projet exemple de conversion AFP en PDF

1. Créez un dossier `input2` dans le répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\Dev`
2. Copiez le fichier `purchaseorder.afp` du répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\data\samples` dans le dossier `input2`.
3. Lisez le fichier journal dans Control Center pour vérifier que le fichier a été traité.
4. Vérifiez que cinq fichiers nommés `nnnnnnnnn.pdf` figurent dans le répertoire `C:\ManagementGateway\1.0\root\applications\`<*nom application*>`\Dev\output` où `nnnnnnn` correspond au numéro du bon de commande dans les données d'entrée.

Configurations du Projet exemple

Configuration Exemple 1

La configuration Exemple 1 correspond à un cas d'utilisation de base avec un post-traitement synchrone, incluant le tri des documents et le marquage OMR.

Vous pouvez connecter le Processus au connecteur de sortie **PDF encrypt**, installer le package LCA exemple et configurer le filtre du connecteur pour qu'il pointe vers l'ordinateur sur lequel le package LCA est installé. Ceci illustre la façon dont LiveCycle Production Print ES2 peut se connecter à un serveur LiveCycle, traiter un PDF et renvoyer le PDF à LiveCycle Production Print ES2.

Configuration Exemple 2

La configuration Exemple 2 illustre comment relier des données ASCII à un modèle de formulaire.

1. Copiez le fichier `Invocie.grb` du répertoire samples dans le répertoire input.
2. Vérifiez que le fichier `output.pdf` a été créé dans le répertoire de sortie.

Configuration Processus de relance

Les configurations Exemple 1 et Processus de relance peuvent être utilisées conjointement pour illustrer l'appel distant de LiveCycle Production Print ES2 à partir de LiveCycle ES2.

1. Installez et déployez le package LCA sur un serveur LiveCycle ES2.
2. Appelez le service `DunningNoticeStage` avec le fichier `DunningNotice.xml` comme entrée (disponible dans les ressources). Le fichier XML et le fichier `DunningNotice.xdp` sont envoyés à LiveCycle Production Print ES2.

   Le fichier de données et le modèle seront fusionnés et stockés dans le référentiel pour post-processeur LiveCycle Production Print ES2.
3. Appelez le service *`DunningNoticePrint`* avec le fichier `lcpp.ppq` (fichier de requête du post-processeur) comme entrée. Le fichier PPQ est envoyé à LiveCycle Production Print ES2, qui récupérera la lettre de relance dans le référentiel pour post-processeur, la formatera en tant que fichier PDF et la copiera dans le dossier de sortie sous le nom `output.pdf`.

Configuration Génération d'une lettre de relance

La configuration Génération d'une lettre de relance illustre la façon dont LiveCycle Production Print ES2 peut être appelé avec des données et un modèle.

LiveCycle Production Print ES2 fusionnera les données et le modèle et renverra les résultats sous la forme d'un fichier AFP à LiveCycle ES2.

1. Appelez le service `DunningNoticeGenerate` avec le fichier `Dunning Notice.xml` comme entrée. Le fichier AFP obtenu sera copié dans un fichier sur le serveur LiveCycle (par défaut, `c:\result.afp`).

Configuration Exemple 5 Conversion AFP en PDF

La configuration Exemple 5 illustre la façon dont des fichiers AFP peuvent être convertis en fichiers PDF via l'outil PreformatIN et des fonds de page dynamiques dans le Processus PageOUT.

L'Evénement PreformatIN lira le fichier `PurchaseOrder.afp` au moyen d'un filtre AFP2LXF et mettra les identifiants en correspondance afin d'identifier le type de document. Le Processus PageOUT appliquera les pages LXF générées dynamiquement à une page logique. Le résultat sera un fichier PDF généré via un pilote sur le connecteur de sortie.

1. Appelez la configuration Conversion AFP en PDF en copiant le fichier `purchaseorder.afp` dans le répertoire d'entrée `Watchfolder2`. Cinq fichiers PDF seront alors générés dans le répertoire de sortie, un pour chaque document dans le fichier AFP.

Configuration de la Plate-forme

La Plate-forme du Projet exemple contient des connecteurs d'entrée et de sortie dans le couche physique nommée `Dev`.

Connecteurs d'entrée

`input`	Connecteur d'analyse de répertoires. Le dossier analysé est `\input` (son chemin d'accès est relatif au répertoire de travail d'un Projet déployé).

`DunningNoticeStage`
`DunningNoticePrint`
`DunningNoticeGenerate`
	Connecteurs Service Request. Exposent les configurations de Message en tant que services via la passerelle de services StreamServe.

`WatchFolder2`	Connecteur d'analyse de répertoires. Le répertoire analysé est `.\input2` (chemin relatif au répertoire de travail). Le type du fichier analysé est `*.afp`. Le filtre AFP2LXF est appliqué dans le pipeline d'entrée du connecteur.

Pour afficher les paramètres de connecteur

1. Activez la vue Plate-forme.
2. Cliquez avec le bouton droit de la souris sur le connecteur et sélectionnez **Paramètres**.
3. Cliquez sur la couche logique et la couche physique dans la boîte de dialogue Paramètres pour afficher tous les paramètres du connecteur.

Connecteurs de sortie

`PDF`	Connecteur de type fichier. Le fichier de sortie est défini sur `.\output\purchaseorder.pdf` (chemin d'accès relatif au répertoire de travail d'un Projet déployé). Le pilote de périphérique est paramétré sur `PDF` avec les options par défaut. Le mode de sortie est paramétré sur `Job` afin de conserver toutes les données de sortie dans un seul fichier de sortie.
`PDF encrypt`	Connecteur de type fichier avec les mêmes paramètres que le connecteur `PDF`. Toutefois, le connecteur `PDF encrypt` dispose d'une configuration de filtre qui permet d'appeler à distance un service LiveCycle, pour crypter et protéger par mot de passe les données de sortie avant leur écriture dans un fichier.
`PostProcessing`	Connecteur de type fichier. Le fichier de sortie est défini sur `.\output\output.pdf` (chemin d'accès relatif au répertoire de travail d'un Projet déployé). Le pilote de périphérique est paramétré sur `PDF` avec les options par défaut. Le mode de sortie est paramétré sur `Job` afin de conserver toutes les données de sortie dans un seul fichier de sortie.
`PPRepository`	Connecteur Post-Processor Repository (Référentiel pour Postprocesseur). Enregistre la sortie dans le référentiel de post-traitement incorporé à l'aide de l'alias `lcpp`. Le pilote de périphérique est défini sur `SDR`.
`LC Response`	Connecteur Null pour lequel l'option **Inclure le résultat dans la réponse de service** est définie. Le pilote de périphérique est défini sur `AFP`.
`PDF Bypass`	Similaire au connecteur PDF, mais le mode de sortie est paramétré sur Processus.

Jeu de ressources

Le jeu de ressources par défaut dans le Projet exemple contient trois dossiers, un pour chaque type de Message :

- `Invoice`
- `Purchase Order`
- `Dunning Notice`

Il existe des ressources exemples pour la configuration des Messages :

- `/Invoice/invoice.grb` est un fichier d'impression texte qui sert d'entrée au Message Exemple 2 – Données d'entrée ASCII.
- `/Invoice/invoice.xdp` est le modèle utilisé par le Message Exemple 2 – Données d'entrée ASCII.
- `/Purchase Order/Purchase Order.xml` sert d'entrée au Message Exemple 1 – Données d'entrée XML.
- `/Purchase Order/Purchase Order.xdp` est un modèle de formulaire prédéfini à utiliser avec le Message Exemple 1 – Données d'entrée XML.
- `/Dunning Notice/Dunning notice.xml` sert d'entrée au Message Exemple 3 – Appel par LiveCycle et Exemple 4 – Génération d'une lettre de relance.
- `/AFP2PDF/AFP2PDF` est un pipeline de filtre avec le filtre AFP2LXF configuré.
- `/AFP2PDF/purchaseorder.afp` est un fichier exemple d'entrée au format AFP pour la configuration Exemple 5 Conversion AFP en PDF.

Messages

Exemple 1 – Données d'entrée XML (modèle de formulaire existant)

Ce Message utilise un modèle statique (`purchase order.xdp`) qui accepte un fichier de données XML comme entrée.

Un Evénement et un Processus sont configurés pour le Message. L'Evénement utilise un identifiant pour reconnaître le type de fichier de données pour le Message. L'identifiant est paramétré sur `/batch` pour qu'il corresponde au nœud racine du fichier de données `Purchase Order.xml`. Le Processus crée une liaison sur un modèle de formulaire lorsque les données d'entrée correspondent à l'identifiant de l'Evénement. Le modèle de formulaire `Purchase Order.xdp` du jeu de ressources défini est chargé dans le Processus.

Paramètres :

- Le mode d'enregistrement est utilisé avec le déclencheur d'enregistrement `transaction`.
- Le mappage des variables StreamServe est utilisé pour mettre en correspondance l'expression SOM `$record.header.txtPONum` avec la variable StreamServe `$ponum`. La variable est utilisée dans les paramètres Runtime pour trier les formulaires dans le fichier d'entrée batch.

Exemple 2 – Données d'entrée ASCII (modèle créé entièrement par l'utilisateur dans Design Center)

Les données d'entrée pour le modèle de formulaire sont contenues dans un fichier texte. L'Evénement utilise l'outil PageIN pour reconnaître le type du fichier de données et extraire son contenu dans un Message StreamServe.

Le fichier exemple `invoice.grb (` du jeu de ressources) est utilisé pour configurer l'Evénement. Un identifiant est utilisé pour détecter la chaîne `INVOICE` dans un jeu de coordonnées sur la page. L'outil Champ dans PageIN a été utilisé pour extraire des données à partir des coordonnées sur la page et les insérer dans le Message.

Le Processus utilise le paramètre **Ajouter le Message sous la forme d'une connexion de données dans la Vue des données** (voir *Boîte de dialogue Paramètres* à la page 84) pour présenter le contenu du Message dans la vue des données LiveCycle Designer. Le contenu de la vue des données a été utilisé pour créer des liaisons vers les champs du modèle de formulaire.

Exemple 3 – Appel par LiveCycle (appel distant de LiveCycle Production Print ES2)

Les données d'entrée de l'Evénement XMLIN sont enregistrées dans le fichier XML `Dunning Notice.xml`. L'identifiant est défini de manière à détecter l'élément `<transaction>`. Le Processus est configuré pour charger un modèle de façon dynamique. La variable `$template` sera automatiquement attribuée à un modèle à partir du connecteur d'entrée.

Exemple 4 – Génération d'une lettre de relance (appel distant de LiveCycle Production Print ES2)

La configuration de ce Message est identique à l'exemple 3. La différence réside dans son utilisation dans la configuration de Runtime.

Exemple 5 - Conversion AFP en PDF

Le configuration de ce Message utilise PreformatIN et PageOUT pour convertir les fichiers AFP en fichiers PDF au moyen du filtre AFP2LXF dans la Plate-forme.

Configurations de Runtime

Job Purchase Order (Bon de commande)

Cette configuration de job utilise le Message Exemple 1. L'Evénement est relié au connecteur d'entrée `WatchFolder`.

La sélection du connecteur de sortie est définie en mode statique et le connecteur de sortie `PostProcessing` est utilisé. Les paramètres du connecteur de sortie `PostProcessing` se basent sur les éléments suivants :

- Marquage OMR dans l'onglet Début du Processus.
- Tri de documents en fonction de la variable `$ponum`. Le tri est défini dans l'ordre décroissant.

Job Invoice (Facture)

Cette configuration de job utilise le Message Exemple 2. L'Evénement est relié au connecteur d'entrée `WatchFolder`.

La sélection du connecteur de sortie est définie en mode statique et le connecteur de sortie `PDF` est utilisé.

Job Dunning notice Stage (Phase de relance)

Cette configuration de job utilise le Message Exemple 3. L'Evénement est relié au connecteur d'entrée `DunningNoticeStage`.

La sélection du connecteur de sortie est définie en mode statique et le connecteur de sortie `PPRepository` est utilisé.

Job Dunning notice Process (Processus de relance)

Il s'agit d'une configuration de type référentiel pour postprocesseur. Le job est relié au connecteur d'entrée `DunningNoticePrint`.

La sélection du connecteur de sortie est définie en mode statique et le connecteur de sortie `PDF` est utilisé.

Job Dunning notice Generate (Génération d'une lettre de relance)

Cette configuration de job utilise le Message Exemple 4. L'Evénement est relié au connecteur d'entrée `DunningNoticeGenerate`.

La sélection du connecteur de sortie est définie en mode statique et le connecteur de sortie `LC Response` est utilisé.

Job AFP2LXF

Ce job définit les paramètres de connecteur de Runtime pour le connecteur PDF Bypass de manière à utiliser une variable pour la génération de noms uniques pour les fichiers PDF.

Prise en charge de l'impression d'étiquettes

LiveCycle Production Print prend en charge les imprimantes d'étiquettes suivantes :

- ZPL II – voir *ZPL II* à la page 68 pour plus de détails.
- Intermec FP/DP – voir *Intermec FP/DP* à la page 70 pour plus de détails.
- Intermec IPL – voir *Intermec IPL* à la page 73 pour plus de détails.
- Printronix PGL/IGP – voir *Printronix PGL/IGP* à la page 76 pour plus de détails.
- TEC – voir *TEC* à la page 78 pour plus de détails.

Limitations dans la prise en charge des imprimantes d'étiquettes

Les langages des imprimantes d'étiquettes comportent des limitations quant à la prise en charge des objets et des propriétés d'objets par rapport à des formats tels que PDF, PCL, AFP, PS etc.

Pour des raisons de performances, LiveCycle Production Print n'offre pas la même prise en charge WYSIWYG que l'aperçu dans Designer ES2. Par exemple :

- Les remplissages avec des couleurs dégradées ne sont pas pixellisés.
- Les styles de lignes, tels que les tirets, ne sont pas pris en charge par tous les langages d'imprimante.

ZPL II

Sortie de texte – ZPL II

Polices à taille variable

Les polices à taille variable suivantes sont prises en charge :

- CG Triumvirate Bold Condensed

C'est la seule police compatible WYSIWYG.

Polices Adobe

Toutes les polices Adobe sont mappées sur la police ci-dessus.

ASCII

Ne prend en charge que la table US ASCII.

Prise en charge de codes à barres – ZPL II

Les codes à barres suivants sont pris en charge :

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Standard
- Code 3 of 9
- Code 49
- Code 93
- Data Matrix
- EAN13
- EAN8
- Logmars
- MSI
- PDF417
- Planet Code
- Plessey
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

Prise en charge RFID – ZPL II

Le code à barres RFID est pris en charge.

Protocole air-interface	EPC Class 1 Generation 2
Valeur du code à barres	Traité comme HEX96.

Prise en charge des objets d'interface utilisateur graphique – ZPL II

Les objets d'interface utilisateur graphique ci-dessous sont pris en charge.

Objet	Commentaire
Lignes verticales	Prises en charge.
Lignes horizontales	Prises en charge.
Lignes diagonales	Prises en charge.
Rectangles	Pris en charge.
Cercles	Pris en charge.
Images	Prises en charge.
Style de ligne	Seul le trait plein est pris en charge.
Rotation	Les rotations de 90, 180 et 270 degrés sont prises en charge.

Intermec FP/DP

Sortie de texte – Intermec FP/DP

Polices à taille variable

Les polices à taille variable suivantes sont prises en charge :

- Century Schoolbook BT
- Dutch 801 Bold BT
- Dutch 801 Roman BT
- Futura Light BT
- Letter Gothic 12 Pitch BT
- Monospace 821 Bold BT
- Monospace 821 BT
- OCR-A BT
- OCR-B 10 Pitch BT
- Prestige 12 Pitch Bold BT
- Swiss 721 Bold BT
- Swiss 721 BT
- Swiss 721 Condensed BT
- Zurich Extra Condensed Bold

Ce sont les seules polices compatibles WYSIWYG.

Polices Adobe

Toutes les polices Adobe sont mappées sur les polices ci-dessus (polices avec des mesures similaires).

ASCII

Ne prend en charge que la table US ASCII.

Prise en charge de codes à barres – Intermec FP/DP

Les codes à barres suivants sont pris en charge :

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Matrix
- Code 2 of 5 Standard
- Code 3 of 9, Code 93
- Code 49, MSI, Planet Code
- Data Matrix
- EAN13
- EAN8
- PDF417
- Plessey
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E, UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

Prise en charge RFID – Intermec FP/DP

Le code à barres RFID est pris en charge.

Protocole air-interface	EPC Class 1 Generation 2
Valeur du code à barres	Traité comme HEX96.

Prise en charge des objets d'interface utilisateur graphique – Intermec FP/DP

Les objets d'interface utilisateur graphique ci-dessous sont pris en charge.

Objet	Commentaire
Lignes verticales	Prises en charge.
Lignes horizontales	Prises en charge.
Lignes diagonales	Non prises en charge.
Rectangles	Pris en charge.
Cercles	Non pris en charge.
Images	Prises en charge.
Style de ligne	Seul le trait plein est pris en charge.
Rotation	Les rotations de 90, 180 et 270 degrés sont prises en charge.

Intermec IPL

Sortie de texte – Intermec IPL

Polices à taille variable

Les polices à taille variable suivantes sont prises en charge :

- Century Schoolbook BT
- Dutch 801 Roman BT
- Letter Gothic 12 Pitch BT
- Monospace 821 BT
- OCR-B 10 Pitch BT
- Swiss 721 Bold BT
- Swiss 721 Condensed BT
- Dutch 801 Bold BT
- Futura Light BT
- Monospace 821 Bold BT
- OCR-A BT
- Prestige 12 Pitch Bold BT
- Swiss 721 BT
- Zurich Extra Condensed Bold

Ce sont les seules polices compatibles WYSIWYG.

Polices Adobe

Toutes les polices Adobe sont mappées sur les polices ci-dessus (polices avec des mesures similaires).

ASCII

Ne prend en charge que la table US ASCII.

Prise en charge de codes à barres – Intermec IPL

Les codes à barres suivants sont pris en charge :

- Aztec
- Codabar
- Code 11
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 3 of 9
- Code 49
- Code 93
- Data Matrix
- EAN13
- EAN8
- PDF417
- Planet Code
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- RSS14 Truncated
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5
-

Prise en charge RFID – Intermec IPL

Le code à barres RFID est pris en charge.

Protocole air-interface	EPC Class 1 Generation 2
Valeur du code à barres	Traité comme HEX96.

Prise en charge des objets d'interface utilisateur graphique – Intermec IPL

Les objets d'interface utilisateur graphique ci-dessous sont pris en charge.

Objet	Commentaire
Lignes verticales	Prises en charge.
Lignes horizontales	Prises en charge.
Lignes diagonales	Non prises en charge.
Rectangles	Pris en charge.
Cercles	Non pris en charge.
Images	Prises en charge.
Style de ligne	Seul le trait plein est pris en charge.
Rotation	Les rotations de 90, 180 et 270 degrés sont prises en charge.

Printronix PGL/IGP

Sortie de texte – Printronix PGL/IGP

Polices à taille variable

Les polices à taille variable suivantes sont prises en charge :

- CG Triumvirate Bold Condensed
- Courier Bold
- Letter Gothic Bold

Ce sont les seules polices compatibles WYSIWYG.

Polices Adobe

Toutes les polices Adobe sont mappées sur les polices ci-dessus (polices avec des mesures similaires).

ASCII

Ne prend en charge que la table US ASCII.

Prise en charge de codes à barres – Printronix PGL/IGP

Les codes à barres suivants sont pris en charge :

- AUSPOST Custom 2
- AUSPOST Custom 3
- AUSPOST Replay Paid
- AUSPOST Standard
- Codabar
- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Matrix
- Code 3 of 9
- Code 93
- Data Matrix
- EAN13
- EAN8
- MSI
- PDF417
- Planet Code
- Plessey
- UK/Royal Mail RM4SCCC
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

Prise en charge RFID – Printronix PGL/IGP

Le code à barres RFID est pris en charge.

Protocole air-interface	EPC Class 1 Generation 2
Valeur du code à barres	Traité comme HEX96.

Prise en charge des objets d'interface utilisateur graphique – Printronix PGL/IGP

Les objets d'interface utilisateur graphique ci-dessous sont pris en charge.

Objet	**Commentaire**
Lignes verticales	Prises en charge.
Lignes horizontales	Prises en charge.
Lignes diagonales	Prises en charge.
Rectangles	Pris en charge.
Cercles	Pris en charge.
Images	Prises en charge.
Style de ligne	Seul le trait plein est pris en charge.
Rotation	Les rotations de 90, 180 et 270 degrés sont prises en charge.

TEC

Sortie de texte – TEC

Polices bitmap

Les polices bitmap suivantes sont prises en charge :

- Courier
- Helvetica
- Letter Gothic
- OCR-A
- OCR-B
- Presentation
- Prestige Elite
- Times New Roman

Ce sont les seules polices compatibles WYSIWYG.

Polices Adobe

Toutes les polices Adobe sont mappées sur les polices ci-dessus (polices avec des mesures similaires).

ASCII

Ne prend en charge que la table US ASCII.

Prise en charge de codes à barres – TEC

Les codes à barres suivants sont pris en charge :

- Code 128
- Code 2 of 5 Industrial
- Code 2 of 5 Interleaved
- Code 2 of 5 Matrix
- Code 3 of 9
- Code 93
- Data Matrix
- EAN13
- EAN8
- MSI
- PDF417
- QR Code
- RSS14
- RSS14 Expanded
- RSS14 Limited
- RSS14 Stacked
- RSS14 Stacked Omnidirectional
- UK/Royal Mail RM4SCCC
- UPC-A
- UPC-E
- UPS Maxicode
- US Postal DPBC
- US Postal Standard
- US Postal Zip-5

Prise en charge RFID – TEC

RFID n'est pas implémenté.

Prise en charge des objets d'interface utilisateur graphique – TEC

Les objets d'interface utilisateur graphique ci-dessous sont pris en charge.

Objet	Commentaire
Lignes verticales	Prises en charge.
Lignes horizontales	Prises en charge.
Lignes diagonales	Prises en charge.
Rectangles	Pris en charge.
Cercles	Pris en charge.
Images	Prises en charge.
Style de ligne	Seul le trait plein est pris en charge.

Objet	Commentaire
Rotation	Les rotations de 90, 180 et 270 degrés sont prises en charge.

Interface utilisateur

Dans ce chapitre :

Outil Processus LiveCycle Designer ES2

La fenêtre de démarrage de l'outil Processus apparaît lorsque vous ouvrez un Processus Adobe LiveCycle Designer ES2.

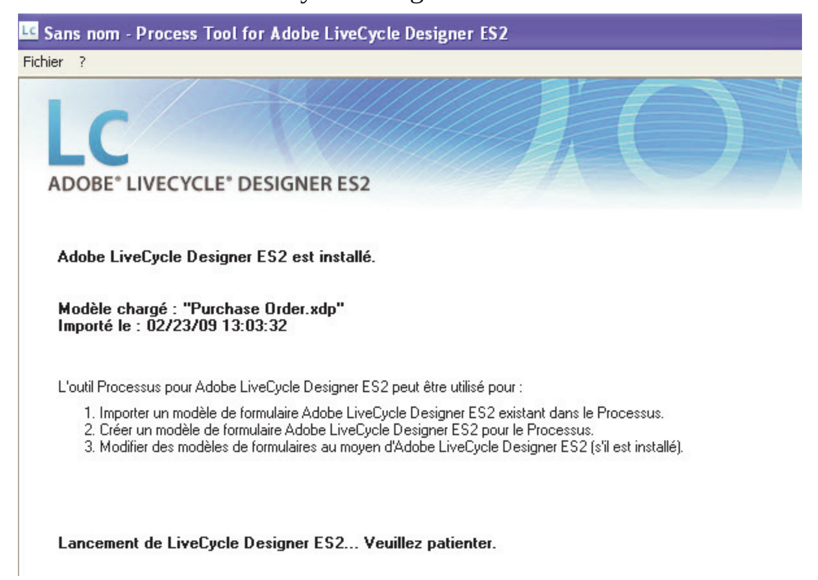


Figure 5 Fenêtre de démarrage de l'outil Processus pour Adobe LiveCycle Designer ES2

Si un modèle est chargé, son nom ainsi que la date à laquelle il a été importé sont affichés.

Si LiveCycle Designer ES2 :

- est installé et si vous avez choisi de charger l'interface utilisateur de LiveCycle Designer au démarrage, cette dernière est lancée.
- n'est pas installé, vous accédez aux commandes du menu **Fichier** dans cette fenêtre pour définir les paramètres et sélectionner le modèle.

Dans ce chapitre :

Commandes du menu Fichier

Ouvrir un modèle	Ouvre la boîte de dialogue Sélection du modèle qui permet de choisir un modèle (voir *Boîte de dialogue Sélection du modèle*).
Enregistrer	Enregistre les modifications effectuées.

Importer un fichier LiveCycle Archive	Ouvre la boîte de dialogue Sélectionner le fichier LCA à importer, dans laquelle vous sélectionnez le fichier LiveCycle Archive (LCA) à importer. Le fichier LCA contient un package complet composé d'un modèle XDP principal et de ses dépendances. Pour pouvoir être importé dans l'outil Processus, le fichier LCA doit avoir été préalablement importé dans Design Center comme type de ressource Exemple.
Exporter le schéma de Message	Crée un fichier de schéma XML contenant le Message StreamServe.
Exporter le fichier XML d'aperçu	Crée un fichier XML qui contient la structure du Message StreamServe, avec des données exemples pour chaque champ.
Extraire les ressources dans un fichier	Ouvre la boîte de dialogue Sélectionner la ou les ressources à extraire, dans laquelle vous pouvez naviguer jusqu'au jeu de ressources et sélectionner une ou plusieurs ressources à extraire dans un fichier. Les ressources extraites sont stockées dans le répertoire de travail actif de l'outil Processus, dans la même structure que dans le jeu de ressources, auquel LiveCycle Designer peut accéder.
Paramètres	Ouvre la boîte de dialogue Paramètres (voir *Boîte de dialogue Paramètres*).
Quitter	Ferme LiveCycle Designer. Si des composants ont été modifiés sans être enregistrés, une boîte de dialogue s'ouvre pour vous demander si vous voulez les enregistrer.

Boîte de dialogue Paramètres

Vous pouvez accéder à la boîte de dialogue Paramètres pour définir des paramètres par défaut et mapper des expressions SOM avec des variables StreamServe.

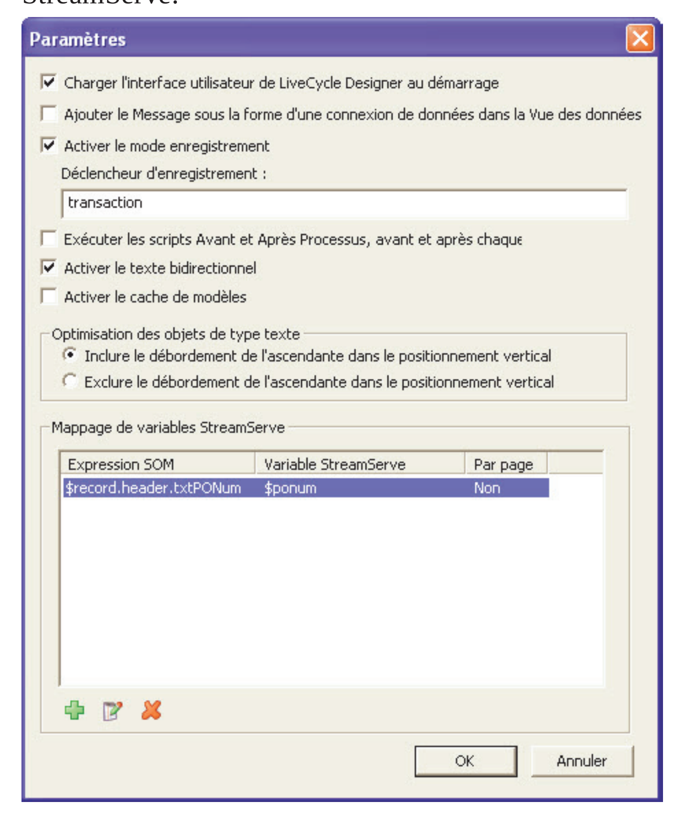


Figure 6 Boîte de dialogue Paramètres

Charger l'interface utilisateur de LiveCycle Designer au démarrage

Sélectionnez cette option si vous souhaitez exécuter l'interface utilisateur de LiveCycle Designer. L'interface s'ouvre automatiquement lorsque vous lancez l'outil Processus.

Note : Cette option n'est disponible que si vous avez installé LiveCycle Designer ES2.

Ajouter le Message sous la forme d'une connexion de données dans la Vue des données

Sélectionnez cette option si vous créez un modèle en définissant tout vous-même. Le Message StreamServe est ajouté à LiveCycle Designer sous la forme d'une connexion de données.

Activer le mode enregistrement

Le Processus peut fonctionner dans deux modes : le mode enregistrement et le mode sans enregistrement. Dans le mode enregistrement, le document de données sera traité comme une séquence d'enregistrements. Dans le cas le plus simple, chaque enregistrement est chargé, traité et déchargé avant que l'enregistrement suivant ne soit à son tour chargé. Le mode enregistrement est uniquement destiné à réduire la consommation de ressources (cycles de mémoire et processeur) en cas de traitement de gros documents de données.

Toutes les tâches qui peuvent être effectuées en mode enregistrement peuvent également l'être en mode sans enregistrement, dans la mesure où il y a suffisamment de ressources disponibles.

Déclencheur d'enregistrement

Ce champ est activé si l'option **Activer le mode enregistrement** est sélectionnée. Il indique comment fragmenter les données d'entrée en plusieurs enregistrements en définissant une balise ou un niveau XML (un nombre positif).

Exemple 5 *Déclencheur d'enregistrement avec la structure de données d'entrée suivante*

```
 <A>
  <B>
    <Données1>1</Données1>
    <Données2>2</Données2>
  </B>
  <C>
    <Données1>3</Données1>
    <Données2>4</Données2>
  </C>
  <B>
    <Données1>5</Données1>
    <Données2>6</Données2>
  </B>
</A>
```

Si vous paramétrez le **Déclencheur d'enregistrement** sur :

- `1`, trois enregistrements sont effectués.
- Balise XML `B`, deux enregistrements sont effectués (ceux avec la balise `B`). L'enregistrement associé à la balise `C` est ignoré.

  Lorsque vous utilisez une balise XML comme déclencheur d'enregistrement, une recherche est exécutée selon l'algorithme de parcours en profondeur (DFS). La première balise XML trouvée est ainsi utilisée avec son niveau pour la suite de la recherche. Seuls sont pris en compte les enregistrements qui sont à la fois balisés avec la balise XML définie et qui se trouvent sur le même niveau que le premier enregistrement trouvé.

Exécuter les scripts Avant et Après Processus, avant et après chaque enregistrement

Ce champ est activé si l'option **Activer le mode enregistrement** est sélectionnée.

Si vous sélectionnez cette option, les scripts Avant et Après Processus sont exécutés avant et après chaque enregistrement. Si l'option n'est pas sélectionnée, les scripts Avant et Après Processus sont exécutés avant et après chaque Processus.

Pour les Projets créés dans LiveCycle Production Print ES2, cette fonctionnalité est activée par défaut.

Pour les Projets créés dans des versions antérieures à LiveCycle Production Print ES2, c'est le paramétrage avec les scripts par Processus qui est utilisé par défaut.

Voir *Utilisation de scripts Avant et Après Processus* à la page 104.

Activer le texte bidirectionnel

Sélectionnez cette option pour activer le texte bidirectionnel. Le texte bidirectionnel peut servir à l'affichage d'un texte en arabe ou en hébreu. L'option **Activer le texte bidirectionnel** est désactivée par défaut.

Activer le cache de modèles

Sélectionnez cette option pour activer la mise en mémoire cache des modèles. Il est ainsi possible d'améliorer sensiblement les performances si vous utilisez des modèles dynamiques.

Voir *Mise en mémoire cache de modèles* à la page 12.

Optimisation des objets de type texte

La valeur de l'ascendante d'une police correspond à la partie d'un caractère qui se situe au-dessus de la ligne de base. Cette valeur est en général identique pour tous les caractères d'une police. Toutefois, si vous utilisez des caractères spéciaux, comme les caractères suédois Ä et Å, la valeur de l'ascendante risque d'être supérieure pour la police donnée, d'où le débordement de l'ascendante. Pour éviter que des lignes de texte ne se chevauchent, la valeur de débordement de l'ascendante peut être ajoutée à la hauteur de la ligne de texte.

- **Inclure le débordement de l'ascendante dans le positionnement vertical**
  Le débordement de l'ascendante est calculé et sa valeur est ajoutée à la hauteur de la ligne de texte pour la police concernée. Le traitement est moins rapide car chaque caractère de la police est pris en compte. Ne sélectionnez pas cette option si vous n'utilisez pas de caractères spéciaux. L'option **Inclure le débordement de l'ascendante dans le positionnement vertical** est activée par défaut.
- **Exclure le débordement de l'ascendante dans le positionnement vertical**
  Le débordement de l'ascendante n'est pas pris en compte. Si vous utilisez des caractères spéciaux, les lignes de texte risquent de se chevaucher. Sélectionnez cette option si vous privilégiez un traitement rapide.

Mappage de variables StreamServe

La liste établit une correspondance entre une **expression SOM** et une **variable StreamServe**.

Par page indique si le mappage se fait par document ou par page.

`Oui`	Le mappage se fait une seule fois avant la génération de chaque page.
`Non`	Le mappage se fait une seule fois avant la génération de chaque document.

Le mappage par page est défini lors de l'édition du mappage des variables dans la boîte de dialogue Mappage de variables, au niveau de l'option **Evaluer la variable pour chaque page**.

Voir *Extraction de la valeur d'une expression SOM dans une variable StreamServe* à la page 100.

Boîte de dialogue Sélection du modèle

Dans la boîte de dialogue Sélection du modèle, sélectionnez un modèle et entrez les informations de connexion correspondantes.

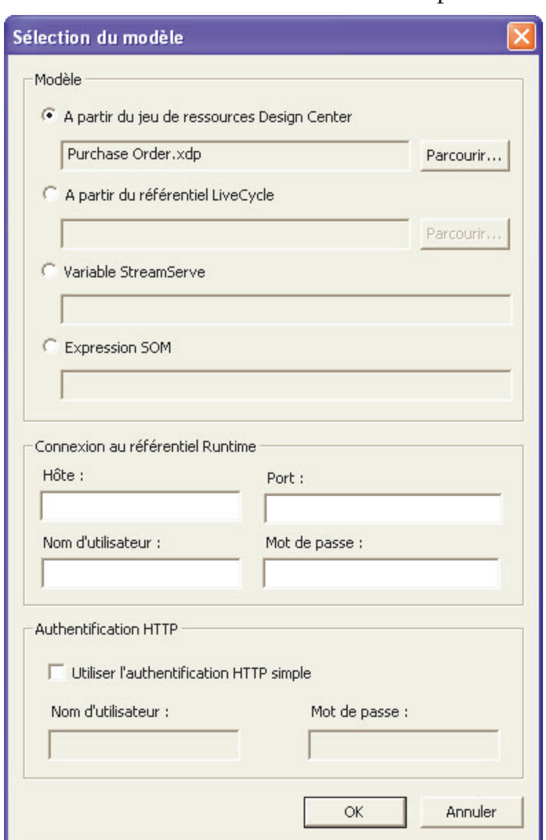


Figure 7 Boîte de dialogue Sélection du modèle

Les paramètres sont décrits dans le tableau ci-dessous :

Paramètres	Description
A partir du jeu de ressources Design Center	Permet d'accéder à un modèle contenu dans un jeu de ressources et de le sélectionner. Le modèle sélectionné sera associé de façon statique au Processus.
A partir du référentiel LiveCycle	Permet d'accéder au modèle à partir du référentiel indiqué et de le sélectionner. Les informations de connexion au référentiel et les identifiants de connexion figurant dans la zone **Connexion au référentiel Runtime** ci-dessous seront utilisés. Le modèle sélectionné sera chargé à partir du référentiel lors de l'exécution, au démarrage du processus.
Variable StreamServe	Permet de spécifier une variable StreamServe qui pointe vers un modèle. Voir *Utilisation d'une variable StreamServe pour charger un modèle* à la page 24.
Expression SOM	Permet de spécifier une expression SOM dans le DOM de données qui pointe vers un modèle. Par exemple, `$record.templateLocation` Voir *Utilisation d'une expression SOM pour charger un modèle* à la page 26.
Connexion au référentiel Runtime	Permet de définir le profil de connexion du référentiel LiveCycle.
Authentification HTTP	Si vous utilisez des URI HTTP, vous pouvez sélectionner l'option **Utiliser l'authentification HTTP simple**. Les **nom d'utilisateur** et **mot de passe** entrés serviront d'identifiants de connexion.

Design Center

Ce chapitre décrit les commandes Design Center propres à LiveCycle Production Print ES2.

Dans ce chapitre :

Commandes du menu Outils

Le menu Outils contient les commandes décrites dans le tableau ci-dessous..

Sélectionner une connexion au référentiel Adobe LiveCycle	Ouvre la boîte de dialogue Sélectionner une connexion au référentiel Adobe LiveCycle, dans laquelle vous pouvez gérer les connexions au référentiel LiveCycle ES2 (voir *Boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle*).

Commandes du menu Ressources

Le menu Ressources est accessible lorsque la vue Jeu de ressources est active. Il contient les commandes décrites dans le tableau ci-dessous.

Importer du référentiel Adobe LiveCycle	Ouvre le navigateur Sélectionner une ressource. Permet d'accéder à une ressource dans le référentiel LiveCycle ES2 et de la sélectionner. Une fois la ressource sélectionnée, la boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle s'ouvre (voir *Boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle*).
Mettre à jour toutes les ressources Adobe LiveCycle	Ouvre la boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle (voir *Boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle*).

Boîtes de dialogue

Boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle

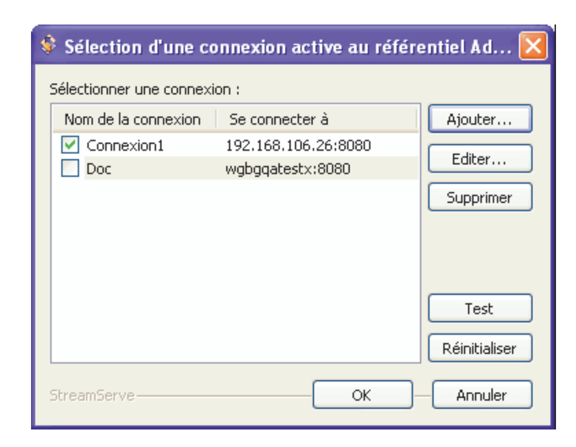


Figure 8 Boîte de dialogue Sélection d'une connexion active au référentiel Adobe LiveCycle

Cette boîte de dialogue affiche la liste de toutes les connexions disponibles vers les référentiels LiveCycle ES2.

Note : Vous pouvez définir des connexions à plusieurs référentiels, mais vous ne pouvez créer et mettre à jour des ressources qu'à partir d'un seul référentiel à la fois.

La connexion sélectionnée est la connexion active.

Paramètre	Description
Sélectionner une connexion :	Toutes les connexions définies vers les référentiels LiveCycle ES2. • **Nom de la connexion** – Nom de la connexion. • **Se connecter à** – Nom ou adresse IP de l'hôte connecté et numéro du port.
Ajouter	Ajoute une nouvelle connexion.
Editer	Permet de modifier une connexion existante.
Supprimer	Supprime la connexion sélectionnée (en surbrillance).
Test	Vérifie si la connexion sélectionnée peut être établie correctement.

Paramètre	Description
Réinitialiser les identifiants de connexion	Efface vos identifiants de connexion au référentiel Adobe LiveCycle. Vous utilisez ce paramètre si le nom d'utilisateur ou le mot de passe ont été modifiés sur le serveur, ou si vous vous êtes trompé en saisissant votre mot de passe.

Boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle

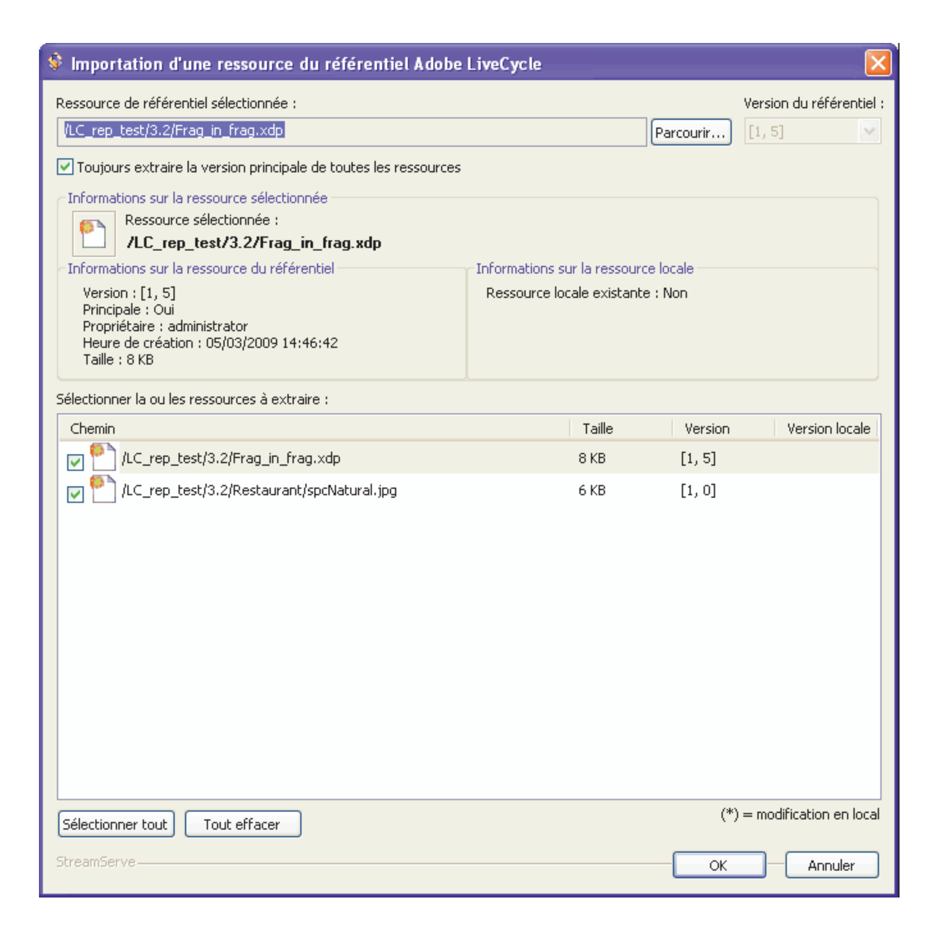


Figure 9　Boîte de dialogue Importation d'une ressource du référentiel Adobe LiveCycle

Voir *Icônes utilisées dans le référentiel LiveCycle ES2* à la page 97 pour plus d'informations sur les icônes de document utilisées dans cette boîte de dialogue.

Les paramètres sont décrits dans le tableau ci-dessous.

Paramètre	Description
Ressource de référentiel sélectionnée	Affiche le chemin et le nom du fichier de la ressource sélectionnée.
Version du référentiel	Version de la ressource dans le référentiel. Vous pouvez sélectionner les versions disponibles dans la liste déroulante.
Toujours extraire la version principale de toutes les ressources	Importe la toute dernière version de la ressource et ses dépendances. Cette option est présélectionnée.
Informations sur la ressource du référentiel	Affiche des informations sur la ressource sélectionnée (en surbrillance dans la liste) dans le référentiel : **Version** – Numéro de version de la ressource sélectionnée. **Principale** – Oui/non Oui si la version actuelle est la plus récente. Non si ce n'est pas le cas. **Propriétaire** – Nom d'utilisateur de l'auteur. **Heure de création** – Date et heure de création de la version de la ressource sélectionnée. **Taille** – Taille du fichier de ressource (Ko). Si la ressource a été supprimée du référentiel, cette information est également affichée.
Informations sur la ressource locale	Affiche des informations sur la ressource sélectionnée (en surbrillance) dans le jeu de ressources locales : **Ressource locale existante** – Oui/non. Oui si la ressource existe déjà en local dans le jeu de ressources Design Center. **Version locale** – Numéro de la version de la ressource stockée en local dans le jeu de ressources Design Center. **Importée** – Date et heure d'importation et de stockage de la ressource dans le jeu de ressources Design Center. **Modifiée en local** – Oui/non. Oui si la ressource a été modifiée en local.

Paramètre	Description
Sélectionner la ou les ressources à extraire	Liste des ressources du référentiel sélectionnées et de leurs dépendances. **Chemin** – Chemin et nom de fichier de la ressource. **Taille** – Taille du fichier de ressource (Ko). **Version** – Numéro de version de la ressource dans le référentiel. **Version locale** – Numéro de la version de la ressource stockée dans le jeu de ressources Design Center.
Sélectionner tout	Sélectionne tous les objets de la liste.
Effacer tout	Décoche toutes les cases (aucun objet sélectionné dans la liste).

Boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle

Cette boîte de dialogue s'affiche si une ressource a été modifiée et doit être mise à jour.

Lorsque la boîte de dialogue s'ouvre, les ressources dont la version dans le jeu de ressources Design Center est antérieure à celle du référentiel sont présélectionnées.

Figure 10 Boîte de dialogue Mise à jour de toutes les ressources du référentiel Adobe LiveCycle

Voir *Icônes utilisées dans le référentiel LiveCycle ES2* à la page 97 pour plus d'informations sur les icônes de document utilisées dans cette boîte de dialogue.

Paramètre	Description
Sélectionner les ressources à mettre à jour	**Chemin** – Chemin et nom de fichier de la ressource. **Taille** – Taille du fichier de ressource. **Version** – Version de la ressource dans le référentiel. **Version locale** – Version de la ressource stockée dans le jeu de ressources Design Center.
Sélectionner tout	Sélectionne tous les objets de la liste.
Effacer tout	Décoche toutes les cases (aucun objet sélectionné dans la liste).

Paramètre	Description
Informations sur la ressource sélectionnée	Informations sur la ressource sélectionnée (en surbrillance).
Informations sur la ressource du référentiel	Affiche des informations sur la ressource du référentiel : **Version** – Version de la ressource sélectionnée. **Principale** – **Oui**/**Non** **Oui** si la version actuelle est la plus récente. **Propriétaire** – Nom d'utilisateur de l'auteur. **Heure de création** – Date et heure de création de la version de la ressource sélectionnée. **Taille** – Taille du fichier de ressource. Si la ressource a été supprimée du référentiel, cette information est également affichée.
Informations sur la ressource locale	**Ressource locale existante** – **Oui**/**Non**. **Oui** si la ressource a été importée dans le jeu de ressources Design Center. **Version locale** – Version de la ressource stockée dans le jeu de ressources Design Center. **Importée** – Date et heure d'importation et de stockage de la ressource dans le jeu de ressources Design Center. **Modifiée en local** – **Oui**/**Non**. **Oui** si la ressource a été modifiée en local.

Boîte de dialogue Paramètres de Processus de Runtime - onglet Général

Dans la boîte de dialogue Paramètres de Processus de Runtime, vous configurez les paramètres de Runtime pour le Processus sélectionné.

L'onglet **Général** vous permet de définir les paramètres généraux du Processus de Runtime.

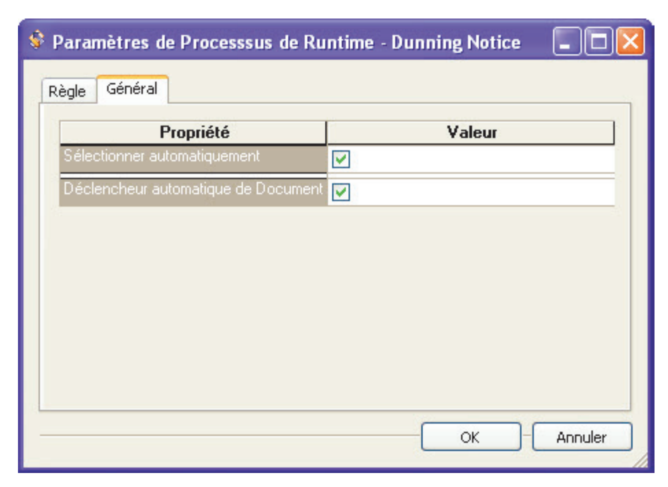


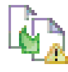
Figure 11 Paramètres de Processus de Runtime

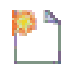
Paramètre	Description
Sélectionner automatiquement	Voir la documentation standard sur *StreamServe Design Center*.
Déclencheur automatique de Document	Si vous sélectionnez ce paramètre, la **variable du déclencheur de Document** (définie dans la boîte de dialogue Paramètres du Connecteur de sortie de Runtime, onglet Déclencheur de Document) est désactivée. Cela signifie que chaque enregistrement est automatiquement mappé avec un document. Il n'y aura pas de regroupement des données de sortie en documents logiques. Ce paramètre est sélectionné par défaut pour les Projets mis à niveau à partir de versions antérieures à LiveCycle Designer ES2. Il n'est pas sélectionné par défaut pour les nouveaux Processus LiveCycle Designer ES2.

Le paramètre **Déclencheur automatique de Document** est également disponible pour les Processus StoryTeller. Chaque Processus sera ainsi automatiquement mappé avec un document.

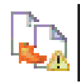
Icônes utilisées dans le référentiel LiveCycle ES2

Les boîtes de dialogue contiennent un certain nombre d'icônes de document :


La version de la ressource stockée dans le référentiel LiveCycle ES2 est postérieure à celle de la ressource locale (dans le jeu de ressources Design Center). L'importation d'une ressource remplacera la ressource


La ressource n'existe pas dans le jeu de ressources Design Center. Elle sera créée si elle est importée.


La version de la ressource dans le jeu de ressources Design Center local est identique à celle de la ressource dans le référentiel LiveCycle ES2. Importation inutile.


La version de la ressource locale (dans le jeu de ressources Design Center) est postérieure à celle de la ressource dans le référentiel LiveCycle ES2. L'importation d'une ressource remplacera la ressource


La ressource a été supprimée du référentiel LiveCycle ES2, mais elle existe toujours en local.

Scripts

Dans ce chapitre :

Extraction de la valeur d'une expression SOM dans une variable StreamServe

Vous pouvez mapper les valeurs d'une expression SOM avec une variable StreamServe.

Il est ainsi possible d'utiliser les données de LiveCycle Designer, pour contrôler le post-traitement ou définir des options de pilote, par exemple, en fonction des données d'entrée, de modèle, de formulaire, de présentation ou de script.

Ce mappage se fait dans un sens seulement. Vous pouvez extraire des valeurs à partir d'une expression SOM dans une variable StreamServe, mais l'inverse n'est pas possible.

Le mappage peut se faire une seule fois pour chaque document ou une seule fois pour chaque page :

- Par document : le mappage se fait une seule fois pour chaque document, immédiatement après la création de toutes les pages mais avant la génération du document.
- Par page : le mappage se fait une seule fois avant la génération de chaque page. Vous disposez ainsi d'une fonctionnalité plus avancée dans Document Broker et Sheet Layout Editor. Par exemple, vous pouvez utiliser une valeur de niveau page extraite d'une expression SOM pour créer des codes OMR appropriés dans la sortie AFP.

  Le mappage se fait d'abord pour le document entier, puis pour chacune des pages avant la génération.

  Note : Dans le cadre du mappage par page, vous utilisez des expressions SOM relatives pour faire référence à des objets de présentation. Les expressions SOM relatives sont évaluées dans le contexte de page, ce qui permet d'avoir des valeurs différentes pour des pages différentes. Les objets qui ne se trouvent pas dans le contexte de page sont évalués dans le contexte de document ; ils ont les mêmes valeurs pour toutes les pages.

Exemple 6 *Expression SOM et variable StreamServe*

Expression SOM : `$record.header.txtPONum`.

Variable StreamServe : `$ponum`

Exemple 7 *Affectation d'une valeur à une variable StreamServe*

Données d'entrée dans le processeur XFA :

```
<?xml version="1.0" encoding="UTF-8"?>
<batch>
<transaction>
 <header>
```

```
  <txtPONum>1234567890</txtPONum>
  <dtmDate>2004-02-08</dtmDate>
  <txtOrderedByCompanyName>Another Company</
txtOrderedByCompanyName>
  <txtOrderedByAddress>123, Any St.</txtOrderedByAddress>
  …
```

Le mappage de la variable permet d'attribuer la valeur de l'élément `txtPONum` à la variable StreamServe `$ponum` :

Expression SOM : `$record.header.txtPONum`

Variable StreamServe : `$ponum`

Exemple 8 *Mappage par page - expression SOM relative pour les éléments enfants d'une page principale*

Les expressions SOM relatives pour les éléments enfants d'une page principale doivent commencer à partir de l'élément enfant de premier niveau de la page principale.

Une page principale `Page1` contient :

- Une zone de texte `PageNumber`
- Un sous-formulaire `CustomerInfo` avec les champs `CustomerName` et `CustomerID`

L'expression SOM relative pour le champ `PageNumber` est "`PageNumber`".

L'expression SOM relative pour le champ `CustomerName` est "`CustomerInfo.CustomerName`".

Exemple 9 *Mappage par page - expression SOM relative pour les éléments enfants d'un sous-formulaire principal*

Les expressions SOM relatives pour les éléments enfants d'un sous-formulaire principal doivent commencer à partir de l'élément enfant (niveau page) de premier niveau du sous-formulaire principal.

Vous pouvez faire référence à un sous-formulaire principal `Data` en utilisant l'expression SOM absolue "`$form.Data`".

Le sous-formulaire `Data` contient le sous-formulaire enfant `Part1`, qui contient le champ `Header1`.

L'expression SOM relative pour `Header1` est "`Part1.Header1`".

Pour plus d'informations sur la syntaxe des expressions SOM, voir l'aide de LiveCycle Designer : *Scripting > Scripting Using LiveCycle Designer ES2 > Referencing Objects in Calculations and Scripts*.

A propos des expressions SOM

L'expression SOM (Scripting Object Model) XFA est un modèle de référencement des valeurs, propriétés et méthodes d'un modèle DOM (Document Object Model) particulier. Un modèle DOM permet de structurer les objets et les propriétés sous forme hiérarchique. Les expressions SOM XFA facilitent l'accès à ces objets et propriétés grâce à une syntaxe de référence d'objet simple.

La spécification SOM est décrite en détail dans la spécification XFA (XML Forms Architecture) .

Pour savoir quelle est l'expression SOM pour un objet XFA particulier, utilisez la propriété de script `somExpression`. Vous pouvez alors extraire la totalité de l'expression pour l'objet.

Exemple 10 *Exemple ECMAscript*

```
this.rawValue = this.somExpression;
```

Si vous définissez ce script sur l'objet pour lequel vous voulez connaître l'expression SOM, l'expression sera attribuée à la valeur de cet objet et la chaîne sera imprimée dans le document de sortie.

Pour plus d'informations sur les expressions SOM, voir la documentation sur *Adobe LiveCycle Designer ES2*.

Accès à des variables StreamServe dans le processeur XFA

Vous pouvez lire des variables StreamServe directement à partir d'un script dans le processeur XFA. C'est le cas lorsqu'un Message StreamServe est utilisé comme connexion de données pour un modèle (par exemple, l'option **Ajouter le Message sous la forme d'une connexion de données dans la Vue des données** est activée).

Toutes les variables StreamServe créées avant l'exécution du Processus peuvent être lues via la connexion de données à l'aide de la syntaxe suivante :

```
$record.variables.variableid;
```

Exemple 11 *Script StreamServe Avant Processus*

```
$myvar = "myvalue";
//Assigns "myvalue" to a StreamServe variable named "myvar"
```

Exemple 12 *Langage EcmaScript pour l'événement Form Ready dans un champ d'un modèle*

```
this.rawValue = $record.variables.myvar;
//Assigns the value of the StreamServe variable "myvar" to the raw
value of the current object.
```

Note : Il est impossible de lire de cette façon les variables en lecture seule, telles que celles créées par SAP (par exemple, variables d'en-tête RDI) et les agents Lawson. Ces variables doivent être attribuées à des variables de script (en lecture/écriture) avant de pouvoir être lues dans le processeur XFA.

Le positionnement du script et le moment de son exécution déterminent si vous pouvez utiliser cette méthode. L'objet `pageSet` et ses descendants peuvent être attribués en toute sécurité aux variables de script dans l'Evénement `layout:ready` et l'Evénement `prePRint` uniquement.

Utilisation de scripts Avant et Après Processus

Vous pouvez exécuter des scripts Avant et Après Processus, avant et après chaque enregistrement.

L'exécution de scripts avant et/ou après des enregistrements individuels donne un contexte d'exécution des scripts pour les valeurs de variables, extraites par le biais d'expressions SOM. Voir *Mappage de variables StreamServe* à la page 87.

Le mappage d'expressions SOM avec des variables StreamServe se fait pour chaque enregistrement (et même pour chaque page d'un enregistrement si l'option **Evaluer la variable pour chaque page** est sélectionnée). Si cette option

- est activée, le script Avant/Après Processus est appelé pour chaque enregistrement avec ces valeurs.
- est désactivée, seules les valeurs extraites du dernier enregistrement sont disponibles pour le script Après Processus.

Note : Le tout premier script Avant Processus et le tout dernier script Après Processus sont exécutés dans la phase de prétraitement (avant et après le Processus proprement dit), ainsi que dans la phase de traitement. Tous les autres scripts Avant et Après Processus sont exécutés dans la phase de traitement uniquement.

Fonctionnement - ordre d'exécution des scripts

L'ordre d'exécution des scripts Avant et Après Processus durant la phase de traitement avec l'option **Evaluer la variable pour chaque page** activée, est présenté dans la figure ci-dessous.

Ordre des événements (dans le temps)

Fin du Processus

Figure 12 Ordre d'exécution des scripts

1 Le script Avant Processus est exécuté avant l'appel du Processus.

2 Le script Avant Processus est exécuté avant le premier enregistrement.

3 Le script Après Processus est exécuté après le premier enregistrement.
4 Le script Avant Processus est exécuté avant les enregistrements suivants.
5 Le script Après Processus est exécuté après les enregistrements suivants.
6 Le script Avant Processus est exécuté avant le dernier enregistrement.
7 Le script Après Processus est exécuté après le dernier enregistrement.
8 Le script Après Processus est exécuté après la fin du Processus.

Pour activer les scripts Avant et Après Processus

Vous pouvez activer et désactiver cette fonctionnalité dans la boîte de dialogue Paramètres, au moyen de l'option **Exécuter les scripts Avant et Après Processus, avant et après chaque enregistrement**. Voir *Boîte de dialogue Paramètres* à la page 84.

Si vous sélectionnez cette option, les scripts Avant et Après Processus sont exécutés avant et après chaque enregistrement. Si l'option n'est pas sélectionnée, les scripts Avant et Après Processus sont exécutés avant et après chaque Processus.

Pour les Projets créés dans LiveCycle Production Print ES2, cette fonctionnalité est activée par défaut.

Pour les Projets mis à niveau, créés dans des versions antérieures à LiveCycle Production Print ES2, c'est le paramétrage précédent avec les scripts par Processus qui est utilisé par défaut.

Restrictions

Les scripts suivants ne peuvent pas être utilisés dans les scripts Avant et Après Processus s'ils sont exécutés au niveau enregistrement dans la phase de traitement :

- `CallProc`
- `CallBlock`
- `PreProcLog`
- `EndDocument`
- `SetExtJobId`
- `GetJobResourceIndex`
- `AddSortDef`
- `AddSortKeys`
- `ODBCConnect`
- `ODBCDisconnect`
- `SAPCreateFunction`
- `SAPInvokeFunction2`
- `SetDestPath`

Journalisation dans le fichier journal StreamServer

Vous pouvez consigner directement dans le journal StreamServer les messages de journal provenant de scripts dans un modèle XDP.

Syntaxe

`xfa.log.message(`*`Param1`*`, chaîne [, `*`gravité`*`])`

`Param1`	StreamServer ignore toujours ce paramètre.
`chaîne`	Chaîne de texte de message qui sera consignée dans le fichier journal. Le message sera précédé par : "`XFAOUT: Message from XFA template script: `"
`Gravité`	Facultatif. Indique le niveau de gravité du message : `i` – Informations `t` – Informations `w` – Avertissement `f`– Erreur

Description

Cette fonction de script permet de consigner directement dans le journal StreamServer les messages provenant de scripts dans un modèle XDP.

Exemple

Exemple 13 *Exemple d'appel au journal*

```
xfa.log.message(1, "texte à consigner dans le fichier journal",
"i");
```

Le message suivant sera ainsi enregistré dans le journal StreamServe :

```
0902 175411 (0096) 3 XFAOUT: Message from XFA template script: texte
à consigner dans le fichier journal
```

Prise en charge XFA

Le processeur XFA LiveCycle Production Print met en oeuvre un sous-ensemble de la spécification XFA, comme l'indique cette section.

La fonctionnalité relative à la création d'une sortie de document interactive n'est généralement pas prise en charge, excepté pour les propriétés de dessin d'objets rendus de façon statique dans les formats d'impression.

Il en va de même pour la mise en oeuvre des fonctions de script FormCalc et ECMA.

Note : LiveCycle Production Print ES2 est conçu pour atteindre un haut niveau de performance dans le traitement des modèles XFA. Le processeur XFA ne fonctionne pas tout à fait comme les processeurs XFA correspondants dans LiveCycle Output ES2 et LiveCycle Forms ES2. Pour des raisons de performances, il existe des différences au niveau de l'utilisation des contextes de scripts, comme l'explique cette section.

Pour plus d'informations sur les limitations connues, voir les *notes de version sur Adobe LiveCycle Production Print ES2*.

Dans ce chapitre :

Eléments XFA

Dans cette section :

Eléments XFA totalement pris en charge

Les éléments XFA ci-dessous sont totalement pris en charge. Pour plus d'informations sur la dernière spécification XFA, consultez la page Adobe XML Forms Architecture (XFA).

- `<arc>`
- `<area>`
- `<bind>`
- `<boolean>`
- `<checkButton>`
- `<color>`
- `<comb>`
- `<contentArea>`
- `<corner>`
- `<date>`
- `<dateTime>`
- `<defaultUi>`
- `<draw>`
- `<edge>`
- `<extras>`
- `<fill>`
- `<float>`
- `<font>`
- `<format>`
- `<image>`
- `<integer>`
- `<line>`
- `<linear>`
- `<margin>`
- `<occur>`
- `<overflow>`
- `<pattern>`
- `<picture>`
- `<proto>`
- `<radial>`
- `<rectangle>`
- `<solid>`
- `<stipple>`
- `<subformSet>`
- `<time>`
- `<ui>`
- `<variables>`
- `<validate>`

- <hyphenation>

Eléments XFA comportant des attributs non pris en charge

Les éléments XFA ci-dessous sont partiellement pris en charge. Voir la section *Template Reference* dans la spécification Adobe XML Forms Architecture (XFA) pour plus d'informations sur les éléments XFA.

Elément	Attributs non pris en charge
<barcode>	• charEncoding • dataPrep • printCheckDigit • rowColumnRatio • upsMode
<border>	• break
<break>	• bookendLeader • bookendTrailer • after • before
<breakAfter>	• leader • trailer • targetType
<breakBefore>	• leader • trailer • targetType
<button>	• highlight
<calculate>	• override
<caption>	• placement
<choiceList>	• commitOn • open • textEntry
<dateTimeEdit>	• hScrollPolicy • picker

Elément	Attributs non pris en charge
<decimal>	• leadDigits
<event>	• activity
<exData>	• maxLength • rid • transferEncoding
<exclGroup>	• access • accessKey
<field>	• access • accessKey
<imageEdit>	• data
<items>	• ref
<keep>	• intact • next • previous
<medium>	• stock • trayIn • trayOut
<numericEdit>	• hScrollPolicy
<pageArea>	• blankOrNotBlank • initialNumber • numbered • oddOrEven • pagePosition
<pageSet>	• relation
<para>	• preserve
<passwordEdit>	• hScrollPolicy
<script>	• runAt
<setProperty>	• connection

Elément	Attributs non pris en charge
<subform>	• access • allowMacro • layout • restoreState
<template>	• baseProfile
<text>	• rid
<textEdit>	• allowRichText • hScrollPolicy • vScrollPolicy
<value>	• override

Eléments XFA utilisés uniquement pour l'ajout de données dans un PDF balisé

Les éléments XFA ci-dessous ne sont utilisés que pour ajouter des données à un PDF balisé. Voir la section *Template Reference* dans la spécification Adobe XML Forms Architecture (XFA) pour plus d'informations.

- `<assist>`
- `<desc>`
- `<speak>`
- `<toolTip>`
- `<traversal>`
- `<traverse>`

Eléments XFA non pris en charge

Les éléments XFA ci-dessous ne sont pas pris en charge.

- <appearanceFilter>
- <certificates>
- <digestMethod>
- <encoding>
- <encrypt>
- <execute>
- <handler>
- <keyUsage>
- <manifest>
- <message>
- <oids>
- <reasons>
- <signData>
- <signing>
- <subjectDNs>
- <timeStamp>
- <bindItems>
- <certificate>
- <connect>
- <digestMethods>
- <encodings>
- <exObject>
- <filter>
- <issuers>
- <lockDocument>
- <mdp>
- <oid>
- <reason>
- <ref>
- <signature>
- <subjectDN>
- <submit>

Fonctions de script XFA

Dans cette section :

Fonctions de script prises en charge

Les fonctions de script répertoriées dans le tableau ci-dessous sont totalement ou partiellement prises en charge. Si une fonction n'apparaît pas dans le tableau, c'est qu'elle n'est pas mise en oeuvre.

- Les fonctions totalement prises en charge sont présentées dans le *manuel de référence des fonctions de script LiveCycle*.
- Les fonctions partiellement prises en charge sont présentées dans le *manuel de référence des fonctions de script LiveCycle*, ainsi que les exceptions mentionnées dans le tableau ci-dessous.

Fonction	Commentaires
`absPage`	Prise en charge totale
`absPageCount`	Prise en charge totale
`absPageSpan`	Prise en charge totale
`addInstance`	• Le paramètre booléen `param` n'est pas pris en charge. • La nouvelle instance n'est pas fusionnée avec le DOM de données. • Cette fonction échoue si la valeur de l'attribut `maxOccurs` d'un sous-formulaire, dans un contexte donné, est dépassée lorsque le sous-formulaire est ajouté.
`addItem`	Prise en charge totale
`assignNode`	• Cette fonction ne s'applique qu'aux noeuds du DOM de formulaire ou du DOM de présentation. • Le paramètre `param3` n'est pas pris en charge. • Il est impossible de créer de nouveaux noeuds. • Cette fonction essaie toujours d'attribuer la valeur à un noeud existant, sinon elle échoue.

Fonction	Commentaires
boundItem	Prise en charge totale
clearItems	Prise en charge totale
clone	Cette fonction ne s'applique qu'aux noeuds du DOM de formulaire ou du DOM de présentation.
deleteItem	Prise en charge totale
execCalculate	Cette fonction n'est prise en charge que pour exclGroup, field, form et subform.
execEvent	Prise en charge totale
execInitialize	Cette fonction n'est prise en charge que pour exclGroup, field, form et subform.
getAttribute	Prise en charge totale
getDisplayItem	Prise en charge totale
getElement	Prise en charge totale
getItemState	Prise en charge totale
getSaveItem	Prise en charge totale
h	• Cette fonction n'est prise en charge que pour field, draw, subform et contentArea. • Le paramètre param3 doit être égal à 0. La fonction renvoie sinon la valeur 0.
isCompatibleNS	• Pour les objets du DOM de modèle, du DOM de formulaire et du DOM de présentation, cette fonction renvoie la valeur "true" (vrai) pour les espaces de noms ci-dessous : http://www.xfa.org/schema/xfa-template/ http://www.xfa.org/schema/xfa-template/2.1/ http://www.xfa.org/schema/xfa-template/2.4/ http://www.xfa.org/schema/xfa-template/2.5/ http://www.xfa.org/schema/xfa-template/2.6/ • Pour les objets du DOM de données, cette fonction renvoie la valeur "true" (vrai) pour : http://www.xfa.org/schema/xfa-data/ • La valeur "false" (faux) est renvoyée pour toutes les autres combinaisons espaces de noms/modèle d'objet.
isPropertySpecified	Prise en charge totale
item	Prise en charge totale

Fonction	Commentaires
`message`	Prise en charge totale
`messageBox`	• Fonction mise en oeuvre en JavaScript uniquement. Aucune boîte de message n'est affichée. • Cette méthode renvoie toujours la valeur 2 (annuler).
`namedItem`	Prise en charge totale
`page`	Prise en charge totale
`pageContent`	• L'accès à tout le contenu de la page n'est pas pris en charge (c'est-à-dire valeur "empty" (vide) pour `param2`). • Le paramètre `param3` n'est pas pris en charge. • Cette fonction recherche toujours dans la totalité de la page, y compris `pageArea` et `contentAreas`.
`pageCount`	Prise en charge totale
`pageSpan`	Prise en charge totale
`relayout`	• Fonctionnalité limitée. La présentation actuelle est entièrement terminée, puis un second passage est exécuté. Ce second passage recrée toutes les pages et présente tous les objets en commençant au début des pages. • Les scripts des noeuds de formulaire ne sont pas renvoyés une fois le second passage effectué. Seuls les scripts présents sur les en-têtes et pieds de page de débordement et les objets standard seront exécutés.
`remerge`	Cette fonction exécute une modification complète du DOM de formulaire, mais cette modification n'est effectuée qu'une seule fois. Si la fonction `remerge` est de nouveau appelée lors du second passage pour la fusion du DOM, cela n'aura aucun impact.
`removeInstance`	Prise en charge totale
`resolveNode`	Prise en charge totale
`resolveNodes`	Prise en charge totale
`saveXML`	Cette fonction est mise en oeuvre, mais l'exactitude du document XML renvoyé n'est pas garantie.
`selectedMember`	Prise en charge totale
`setAttribute`	Prise en charge totale
`setElement`	Cette fonction est mise en oeuvre, mais elle ne fonctionne pas correctement. Le remplacement de certains éléments n'est pas toujours répercuté dans la sortie.

Fonction	Commentaires
setItems	Prise en charge totale
setItemState	Prise en charge totale
sheet	Cette fonction renvoie toujours la même valeur que la valeur page.
sheetCount	Cette fonction renvoie toujours la même valeur que la valeur page.
w	• Cette fonction n'est prise en charge que pour field, draw, subform, contentArea et pageArea. • Le paramètre param3 doit être égal à 0. La fonction renvoie sinon la valeur 0.
x	Le paramètre param3 doit être égal à 0. La fonction renvoie sinon la valeur 0.
y	Le paramètre param3 doit être égal à 0. La fonction renvoie sinon la valeur 0.

Propriétés de script prises en charge

Les propriétés de script répertoriées dans le tableau ci-dessous sont totalement ou partiellement prises en charge. Si une propriété n'apparaît pas dans le tableau, c'est qu'elle n'est pas mise en oeuvre.

- Les propriétés totalement prises en charge sont présentées dans le *manuel de référence des fonctions de script LiveCycle*.
- Les propriétés partiellement prises en charge sont présentées dans le *manuel de référence des fonctions de script LiveCycle*, ainsi que les exceptions mentionnées dans le tableau ci-dessous.

Propriété	Commentaires
#text	Prise en charge totale
{default}	Prise en charge totale
activity	Cette propriété est en lecture seule et ne peut pas être définie à partir des scripts.
after	Prise en charge totale
afterTarget	Lorsque vous définissez cette propriété, seuls les noms des sous-formulaires du modèle actuel peuvent être utilisés.
all	Prise en charge totale
allowNeutral	Prise en charge totale

Propriété	**Commentaires**
`allowRichText`	Prise en charge totale
`anchorType`	Prise en charge totale
`aspect`	Prise en charge totale
`baselineShift`	Prise en charge totale
`before`	Prise en charge totale
`beforeTarget`	Lorsque vous définissez cette propriété, seuls les noms des sous-formulaires du modèle actuel peuvent être utilisés.
`borderColor`	Prise en charge totale
`borderWidth`	Prise en charge totale
`bottomInset`	Prise en charge totale
`break`	Prise en charge totale
`cap`	Prise en charge totale
`charEncoding`	Prise en charge totale
`checksum`	Prise en charge totale
`circular`	Prise en charge totale
`classAll`	Prise en charge totale
`classIndex`	Prise en charge totale
`className`	Prise en charge totale
`colSpan`	Prise en charge totale
`columnWidths`	Prise en charge totale
`contentType`	Cette propriété est prise en charge pour les objets suivants : • `exData` (lecture/écriture) • `image` (lecture/écriture) • `script` (lecture seule).
`count`	Prise en charge totale
`cSpace`	Prise en charge totale
`data`	Prise en charge totale
`dataColumnCount`	Prise en charge totale
`dataLength`	Prise en charge totale
`dataRowCount`	Prise en charge totale

Propriété	Commentaires
editValue	Cette propriété permet de définir ou d'obtenir la valeur rawValue. Aucune distinction n'est effectuée entre les valeurs rawValue et editValue.
endChar	Prise en charge totale
errorCorrectionLevel	Prise en charge totale
excludeAllCaps	La mise en oeuvre de la césure diffère entre LiveCycle et StreamServe. Il n'est donc pas garanti que le texte auquel la césure est appliquée aura le même rendu.
excludeInitialCap	La mise en oeuvre de la césure diffère entre LiveCycle et StreamServe. Il n'est donc pas garanti que le texte auquel la césure est appliquée aura le même rendu.
fillColor	Prise en charge totale
fontColor	Prise en charge totale
fontHorizontalScale	Prise en charge totale
fontVerticalScale	Prise en charge totale
formattedValue	Lorsque vous définissez cette propriété, la propriété rawValue l'est également. Lorsque vous activez cette propriété, la valeur rawValue est formatée avant de revenir au programme d'appel.
fracDigits	Prise en charge totale
h	Prise en charge totale
hAlign	Prise en charge totale
hand	Prise en charge totale
highlight	Prise en charge totale
href	Fonction utilisée uniquement dans les objets image. Elle peut également être définie et lue dans exData, mais n'a aucune signification dans cet objet.
id	Lecture seule
index	Lecture seule
initial	Prise en charge totale
instanceIndex	Lecture seule
intact	Prise en charge totale
inverted	Prise en charge totale
isContainer	Prise en charge totale

Propriété	**Commentaires**
`isNull`	Prise en charge totale
`join`	Prise en charge totale
`kerningMode`	Prise en charge totale
`ladderCount`	Prise en charge totale
`layout`	Prise en charge totale
`leadDigits`	Prise en charge totale
`leader`	Prise en charge pour `breakAfter`, `breakBefore` et `overflow`.
`leftInset`	Prise en charge totale
`length`	Prise en charge totale
`letterSpacing`	Prise en charge totale
`lineHeight`	Prise en charge totale
`lineThrough`	Prise en charge totale
`lineThroughPeriod`	Prise en charge totale
`listen`	Lecture seule
`locale`	Prise en charge totale
`long`	Prise en charge totale
`marginLeft`	Prise en charge totale
`marginRight`	Prise en charge totale
`mark`	Prise en charge totale
`bind`	Lecture seule
`max`	Prise en charge totale
`maxChars`	Prise en charge totale
`maxH`	Prise en charge totale
`maxW`	Prise en charge totale
`min`	Prise en charge totale
`minH`	Prise en charge totale
`minW`	Prise en charge totale
`moduleHeight`	Prise en charge totale
`moduleWidth`	Prise en charge totale

Propriété	**Commentaires**
multiLine	Prise en charge totale
name	Prise en charge totale
next	L'option pageArea n'est pas prise en charge.
nodes	Prise en charge totale
ns	Lecture seule
numberOfCells	Prise en charge totale
numPages	Lecture seule
orientation	Prise en charge totale
overflowLeader	Lecture seule
overflowTarget	Lecture seule
overflowTrailer	Lecture seule
override	Lecture seule. Elle peut être définie, mais n'a aucun impact une fois le modèle chargé.
parent	Lecture seule
parentSubform	Lecture seule
passwordChar	Prise en charge totale
placement	Prise en charge totale
posture	Prise en charge totale
presence	Prise en charge totale
preserve	Prise en charge totale
previous	L'option pageArea n'est pas prise en charge.
printCheckDigit	Prise en charge totale
pushCharacterCount	La mise en oeuvre de la césure diffère entre LiveCycle et StreamServe. Il n'est donc pas garanti que le texte auquel la césure est appliquée aura le même rendu.
radius	Prise en charge totale
radixOffset	Prise en charge totale
rate	Prise en charge totale

Propriété	Commentaires
rawValue	formattedValue et editValue correspondent aux mêmes valeurs que rawValue dans Production Print. Lorsque vous définissez l'une de ces valeurs, toutes auront alors la même valeur. formattedValue sera toutefois formatée une fois encore lors de la sortie. Ce problème est connu.
ref	Lecture seule
relation	Lecture seule
relevant	Prise en charge totale
remainCharacterCount	La mise en oeuvre de la césure diffère entre LiveCycle et StreamServe. Il n'est donc pas garanti que le texte auquel la césure est appliquée aura le même rendu.
reserve	Prise en charge totale
rightInset	Prise en charge totale
rotate	Prise en charge totale
runAt	Traitée en lecture seule. Elle peut être définie, mais n'aura aucun impact.
save	Prise en charge totale
scope	Lecture seule
selectedIndex	Prise en charge totale
shape	Prise en charge totale
short	Prise en charge totale
size	Prise en charge totale
slope	Prise en charge totale
somExpression	Lecture seule
spaceAbove	Prise en charge totale
spaceBelow	Prise en charge totale
startAngle	Prise en charge totale
startChar	Prise en charge totale
startNew	Prise en charge totale
stroke	Prise en charge totale
sweepAngle	Prise en charge totale
tabDefault	Prise en charge totale

Propriété	Commentaires
target	Prise en charge pour breakAfter et breakBefore. Lorsque vous définissez cette propriété, seuls les noms des sous-formulaires du modèle actuel peuvent être utilisés.
targetType	Les options pageEven et pageOdd ne sont pas prises en charge.
textIndent	Prise en charge totale
textLocation	Prise en charge totale
thickness	Prise en charge totale
this	Prise en charge totale
topInset	Prise en charge totale
trailer	Prise en charge pour breakAfter et breakBefore. Lorsque vous définissez cette propriété, seuls les noms des sous-formulaires du modèle actuel peuvent être utilisés.
transferEncoding	Cette fonction n'est prise en charge que pour image.
truncate	Prise en charge totale
type	Cette fonction est prise en charge pour barcode, linear, pattern et radial.
typeface	Cette fonction est prise en charge, mais la police correspondante doit être incluse dans le Projet StreamServe.
underline	Prise en charge totale
underlinePeriod	Prise en charge totale
use	Lecture seule
usehref	Lecture seule
vAlign	Prise en charge totale
value	Pour les objets qui prennent en charge rawValue, value et rawValue renverront le même résultat et définiront la même valeur.
w	Prise en charge totale
weight	Prise en charge totale
wideNarrowRatio	Prise en charge totale
wordCharacterCount	La mise en oeuvre de la césure diffère entre LiveCycle et StreamServe. Il n'est donc pas garanti que le texte auquel la césure est appliquée aura le même rendu.
x	Prise en charge totale

Propriété	Commentaires
`y`	Prise en charge totale

Modèles d'objet de script pris en charge

Les modèles d'objet de script répertoriés dans le tableau ci-dessous sont totalement ou partiellement pris en charge. Voir *Fonctions de script prises en charge* à la page 113 et *Propriétés de script prises en charge* à la page 116 pour plus d'informations sur la prise en charge partielle.

Si un modèle d'objet ne figure pas dans la liste, c'est qu'il n'est pas pris en charge.

- `Data model (modèle de données)`
- `Form model (modèle de formulaire)`
- `Host model (modèle d'hôte)`
- `Layout model (modèle de présentation)`
- `Log model (modèle de journal)`
- `XFA model (modèle XFA)`

Evénements pris en charge

Les événements figurant dans la liste ci-dessous font l'objet d'une prise en charge totale. Les événements totalement pris en charge sont présentés dans le *manuel utilisateur LiveCycle Designer ES2 Scripting Basics*.

Si un événement ne figure pas dans la liste, c'est qu'il n'est pas pris en charge.

- `calculate`
- `form:ready`
- `indexChange`
- `initialize`
- `layout:ready`
- `prePrint`
- `validate`

Considérations relatives aux fonctions de script

Fonctions de script dans le DOM de présentation

Le DOM de présentation est d'abord entièrement créé, puis distribué dans les pages, les zones de contenu, etc. Ensuite, tous les scripts layout-ready et preprint sont exécutés dans l'ordre indiqué dans la spécification XFA 3.1. Si un script layout-ready ou preprint modifie le DOM de présentation et si, suite à cette modification, certains objets basculent sur une autre page, le DOM de présentation est entièrement recréé.

Le recréation du DOM de présentation entraîne la suppression de tous les objets suivants :

- Objets de présentation créés dynamiquement, tels que les caractères de début et de fin
- Objets sur les pages principales

Tous les autres objets de présentation (tels que les champs du DOM de formulaire) restent inchangés, de même que les valeurs de ces objets.

Pour que les valeurs restent correctes, aucun script de présentation d'objet de formulaire n'est exécuté une fois le DOM de présentation recréé. Seuls les scripts attachés aux objets de présentation créés dynamiquement (caractères de début, objets de page principale, etc.) sont exécutés.

Les scripts du DOM de présentation ne sont exécutés qu'une seule fois pour chaque objet, avec toutefois quelques exceptions :

- Les scripts layout-ready et preprint sur le sous-formulaire racine sont tous les deux exécutés la première fois et la seconde fois (le cas échéant) que le DOM de présentation est créé et organisé.
- Les scripts initialize et calculate pour les objets de page principale peuvent être exécutés deux fois.

Vous devez donc être particulièrement attentif lorsque vous utilisez des variables globales et lorsque vous les initialisez, ou lorsque vous accédez à des champs et à d'autres valeurs dans des sous-formulaires créés dynamiquement dans le DOM de formulaire :

- Toute initialisation des variables globales et toute modification apportée en externe (à partir d'autres parties du DOM) aux objets de présentation créés dynamiquement doivent être effectuées dans le sous-formulaire racine ou dans les fonctions de script appelées à partir du sous-formulaire racine.
- Si vous mettez à jour des valeurs dans les objets du DOM de formulaire au sein d'un en-tête de débordement par exemple, il se peut que vous obteniez des résultats inattendus.

  Exemple : `subform.field.x = subform.field.x + 100;` dans un caractère de début entraîne une augmentation de 100 de la valeur `subform.field.x` lorsque le DOM de présentation est créé la première fois. Si le DOM de présentation est créé une nouvelle fois, la valeur 100 est encore ajoutée à `x`.

Variables globales incorporées dans des scripts Java

- Les variables globales incorporées dans des scripts Java dans le DOM de formulaire (par exemple `initialize` ou `form ready`) ne peuvent pas être utilisées dans le DOM de présentation avec les valeurs initialisées. Il y a toutefois une exception à cette règle : les variables globales déclarées dans des objets de script dédiés. Ces variables peuvent être initialisées dans des scripts fu DOM de formulaire et être utilisées dans des scripts du DOM de présentation.

- Il est impossible d'employer dans des scripts form calc et des expressions SOM des variables globales incorporées dans des scripts Java (par exemple, pour transférer des valeurs dans les variables StreamServe). Pour pouvoir être utilisées, ces variables doivent être déclarées en tant que variables de formulaire (c'est-à-dire qu'elles doivent apparaître dans la hiérarchie LiveCycle Designer ES2).

Ordre d'exécution des scripts

L'ordre d'exécution des scripts pour les objets au sein d'un type d'Evénement dans XFA n'est pas défini. N'écrivez pas de scripts qui reposent sur un ordre d'exécution particulier.

Glossaire

Cette section décrit certains termes et concepts utilisés dans ce document.

Control Center StreamServe Control Center permet de gérer et d'administrer des applications StreamServe.

Design Center Design Center est un outil qui permet de configurer des Projets StreamServe.

Dépendances Fragments et images référencés à partir d'un fichier XDP (fichier XDP principal).

Déployer un Projet Lorsque vous déployez un Projet, les fichiers de configuration sont décompressés à partir du fichier d'exportation et stockés dans le répertoire de travail de l'application StreamServer.

Document Un Document (avec un D majuscule) est un groupe de documents réunis par numéro de client ou adresse de livraison, par exemple. La portée d'un Document est définie par un déclencheur de Document.

Evénement Un Evénement détermine comment identifier et extraire des champs à partir de données d'entrée et comment structurer et attribuer des libellés à ces champs.

Exporter un Projet Lorsque vous exportez un Projet, vous générez un fichier de package (`*.export`) qui contient tous les fichiers de configuration requis pour exécuter un Projet. Vous exportez un Projet à partir de Design Center.

LCA Abréviation de LiveCycle Archive.

Outil Processus LiveCycle Designer Application développée par StreamServe qui intègre Adobe LiveCycle Designer ES2 dans StreamServe Design Center.

LiveCycle ES2 Abréviation de Adobe LiveCycle ES2 (Enterprise Suite)

Message

Un Message gère la conversion de données d'entrée dans un format de sortie. Un Projet contient généralement plusieurs Messages, chaque Message correspondant à un type de document particulier (par exemple, un Message pour les factures, un autre pour les commandes, etc.).

Un Message contient des Evénements et des Processus.

PageIN

Outil Evénement qui traite les données d'entrée au format page. Pour plus d'informations sur les autres outils Evénement, voir la documentation sur *StreamServe Design Center*.

PageOUT

Outil Processus qui produit des données de sortie au format page. Pour plus d'informations sur les autres outils Processus, voir la documentation sur *StreamServe Design Center*.

Plate-forme

La Plate-forme permet de configurer les paramètres d'environnement d'un Projet. Par exemple, vous définissez comment vous connecter aux applications sources et recevoir des données d'entrée à partir de ces applications, et comment vous connecter aux périphériques de sortie (imprimantes, fax, etc.) pour leur envoyer des données de sortie.

Processus

Un Processus détermine quels champs doivent être extraits et comment structurer ces champs dans la sortie envoyée à partir de StreamServer.

Outil Processus

Outil Processus pour LiveCycle Designer ES2. Cet outil est développé par StreamServe et intègre Adobe LiveCycle Designer ES2 dans StreamServe Design Center.

Projet

Un Projet StreamServe définit comment les données sont collectées, transformées et envoyées par une solution EDP spécifique.

Ressource

Une ressource est un fichier avec un fichier source incorporé. Par exemple, pour pouvoir utiliser le fichier graphique `logo.gif` dans un Projet, ce fichier doit être incorporé dans une ressource.

Tous les fichiers externes, excepté les dépendances, auxquels vous faites référence lorsque vous configurez un Projet doivent être convertis en ressources.

Jeu de ressources

Ensemble de liens vers des fichiers de ressources.

Runtime	Une configuration de Runtime permet de connecter des Messages à la Plate-forme. Un Projet contient en général plusieurs configurations de Runtime (une configuration de Runtime par Message, par exemple).
StreamIN	Outil Evénement qui traite des données d'entrée structurées en champs ou en enregistrements.
StreamServer	Composant qui gère la collecte et la conversion des données d'entrée et l'envoi des documents de sortie. StreamServer peut exécuter plusieurs applications StreamServer, chacune étant dédiée à un Projet StreamServe en particulier.
XDP	Abréviation de XML Data Package
XFA	Abréviation de XML Forms Architecture
Processeur XFA	Application développée par StreamServe qui permet à l'environnement d'exécution StreamServer de générer des sorties dans différents formats à partir de fichiers de données XML et de modèles de formulaires Adobe LiveCycle Designer ES2.
XMLIN	Outil Evénement qui traite des données d'entrée au format XML.